MP MASTER
MA24041

APPENDIX I

Zm2000 Device Support (Rev 35.0)
(9th February 1988)

When all devices up to and including this revision have been correctly inserted the complete Device Support List for the Zm2000 module should consist of the following pages:

Page 1	Issue 33.0
Page 2	Issue 35.0
Page 3	Issue 35.0
Page 4	Issue 35.0
Page 5	Issue 33.0
Page 6	Issue 31.0
Page 7	Issue 35.0
Page 8	Issue 34.0
Page 9	Issue 35.0
Page 10	Issue 34.0

AMD	Device	Code	Rev
EPROMs	2716	9F42	31.0
	2716B	9FC2	30.0
	2732	9F44	31.0
	2732A	9FC4	31.0
	2732B	9FC3	30.0
	2764	9F4A	27.0
	2764A	9FCA	27.0
	27128	9F4B	27.0
	27128A	9FCB	27.0
	27128AF.+	9FFB	31.0
	27256	9F4C	27.0
	27256F..+	9FFC	31.0
	27C256	9FDC	30.0
	27C256F.+	9FFC	31.0
	27512	9F4D	27.0
	27C512	9FDD	30.0
	27C512F.+	9FFD	31.0
EEPROMs	2817A	9F58	31.0
	2864A	9F5C	31.0
	2864AE	9F5C	31.0
	2864B	9F5D	31.0
	2864BE	9F5D	31.0
	28C256	9F7F	31.0
	9864	9F5A	31.0
Bipolar	27S65..*	991C	31.0
PROMs	27S75..*	991D	31.0
	27S85..*	991E	31.0
	27S08	9801	27.0
	27S09	9801	27.0
	27S18	9801	27.0
	27S19	9801	27.0
	27S10	9802	27.0
	27S11	9802	27.0
	27S20	9802	27.0
	27S21	9802	27.0
	27S12	9803	27.0
	27S13	9803	27.0
	27S32	9806	27.0
	27S33	9806	27.0
	27S184	9808	27.0
	27S185	9808	27.0

AMD Cont..	Device	Code	Rev
	27S28	9812	27.0
	27S29	9812	27.0
	27S40	981A	27.0
	27S41	981A	27.0
	27S290	9823	27.0
	27S291	9823	27.0
	27S280	9827	27.0
	27S281	9827	27.0
	27S51	9829	31.0
	27S25	982A	27.0
	27S35	982B	27.0
	27S37	982B	27.0
	27S45	982D	33.0
	27S47	982D	33.0
	27S27...**	982F	27.0
	27S190	9833	27.0
	27S191	9833	27.0
	27S43	9834	27.0
	27S49	9835	27.0
	27S30	9836	27.0
	27S31	9836	27.0
	27S180	9837	27.0
	27S181	9837	27.0
	27S15...***	9839	27.0

ATMEL	Device	Code	Rev
EPROMs	27HC64L	B7DA	31.0
	27256	B74C	30.0
	27C256	B7DC	30.0
	27HC256	B7DC	30.0
	27HC256L	B7DC	30.0
	27C512	B7DD	30.0
	27C513	B7CD	30.0
EEPROMs	28C04	B770	31.0
	28C04F	B771	31.0
	28C16	B774	31.0
	28C16F	B775	31.0
	28C17	B778	31.0
	28C17F	B779	31.0
	28C64	B77C	31.0
	28C64F	B77D	31.0
	28C256	B77F	31.0

\+ (Flashrite)
* Requires 20-9876 Adaptor
** Requires 20-982F Adaptor
*** Requires 20-1839 Adaptor

CYPRESS	Device	Code	Rev
CMOS PROMs	7C281	2C65	27.0
Series 1	7C282	2C62	27.0
	7C291	2C66	33.0
	7C292	2C63	33.0
	7C225	2C6A	27.0
	7C235	2C6B	27.0
	7C245	2C6D	33.0
CMOS PROMs	7C261	2D25	35.0
Series 2	7C263	2D25	35.0
	7C264	2D35	34.0

EMULATOR	Device	Code	Rev
EPROMs	2716	0E42	31.0
	27C16	0E42	31.0
	2732	0E44	31.0
	27C32	0E44	31.0
	2732A	0EC4	31.0
	27C32H	0E46	31.0
	2764	0E4A	31.0
	2764A	0ECA	31.0
	27C64	0EDA	31.0
	27128	0E4B	31.0
	27128A	0ECB	31.0
	27C128	0EDB	31.0
	27256	0E4C	31.0
	27C256	0EDC	31.0
	57C256	0EDC	31.0
	27512	0E4D	31.0
	27C512	0EDD	31.0
	87C32	0ED4	31.0
	87C64	0EEA	31.0
	87C128	0EEB	31.0
	87C256	0EEC	31.0

EURO-TECHNIQUE /THOMSON	Device	Code	Rev
EPROMs	2716	C742	33.0
	C2716	C7D2	31.0
	C2732	C7D4	31.0
	2764	C74A	30.0
	27C64	C7DA	30.0
	27C256	C7DC	30.0

EXEL	Device	Code	Rev
EEPROMs	2816A	0152	31.0
	2817A	0158	31.0
	2864A	015C	31.0
	2865A	015C	31.0
	46C15	01BA	31.0
	46C16	01BA	31.0
	46C32	01BB	31.0
	48C64	017C	31.0

FAIRCHILD	Device	Code	Rev
BIPOLAR	93417	8802	27.0
PROMs	93427	8802	27.0
Generic 1	93436	8803	27.0
	93446	8803	27.0
	93452	8805	27.0
	93453	8805	27.0
	93514	8808	27.0
	93515	8808	27.0
	93512	8816	27.0
	93513	8816	27.0
	93438	8831	27.0
	93448	8831	27.0
	93450	8832	27.0
	93451	8832	27.0
	93465	8832	27.0
	93466	8832	27.0
	93510	8833	27.0
	93511	8833	27.0
	93564	8835	27.0
	93565	8835	27.0
BIPOLAR	93Z450	8932	27.0
PROMs	93Z451	8932	27.0
Isoplanar-Z	93Z510	8933	27.0
	93Z511	8933	27.0
	93Z564	8935	27.0
	93Z565	8935	27.0

FUJITSU	Device	Code	Rev
EPROMs	2716	AF42	31.0
	2732	AF44	31.0
	2732A	AFC4	31.0
	2764	AF4A	27.0
	27C64	AF4A	27.0
	2764A	AFCA	27.0
	27128	AF4B	27.0
	27128A	AFCB	27.0
	27C128	AFDB	27.0
	27256	AF4C	30.0
	27C256	AFDC	27.0
	27C256A	AFD8	30.0
	27C512	AFDD	27.0
	27C1001.*	AFE1	35.0
EEPROMs	28C64	AF7C	31.0
	28C65	AF7C	31.0
Bipolar	7056	A801	27.0
Proms	7051	A801	27.0
	7057	A802	27.0
	7052	A802	27.0
	7058	A803	27.0
	7053	A803	27.0
	7059	A805	27.0
	7054	A805	27.0
	7060	A832	27.0
	7055	A832	27.0
	7121	A905	27.0
	7122	A905	27.0
	7127	A908	27.0
	7128	A908	27.0
	7123	A912	27.0
	7124	A912	27.0
	7133	A916	27.0
	7134	A916	27.0
	7126	A931	27.0
	7131	A932	27.0
	7132	A932	27.0
	7137	A933	27.0
	7138	A933	27.0
	7141	A934	27.0
	7142	A934	27.0
	7143	A935	27.0
	7144	A935	27.0
	7232	A92B	27.0
	7113	A902	27.0
	7114	A902	27.0
	7115	A903	27.0
	7116	A903	27.0
	7117	A910	27.0
	7118	A910	27.0
	7151	A91A	27.0
	7152	A91A	27.0

GENERAL INSTRUMENTS	Device	Code	Rev
EPROMs	27C64	02DA	30.0
	27HC64	02DA	30.0
	27256	024D	30.0
	27C256	02DC	30.0
	27C512	02DD	30.0
EEPROMs	28C04	0270	31.0
	28C04F	0271	31.0
	2816	0274	31.0
	2816A	0274	31.0
	28C16F	0275	31.0
	28C17	0278	31.0
	28C17A	0278	31.0
	28C17F	0279	31.0
	28C64	027C	31.0
	28C64A	027C	31.0
	28C64F	027D	31.0

HARRIS	Device	Code	Rev
Bipolar	7602	5801	32.0
Proms	7603	5801	32.0
	7610	5802	32.0
	7611	5802	32.0
	7620	5803	32.0
	7621	5803	32.0
	7642	5805	32.0
	7642P	5805	32.0
	7643	5805	32.0
	7643P	5805	32.0
	7684	5808	32.0
	7684P	5808	32.0
	7685	5808	32.0
	7685P	5808	32.0
	7648	5812	32.0
	7649	5812	32.0
	7686	5815	32.0
	7687	5815	32.0
	76164	581A	32.0
	76165	581A	32.0
	7629	5830	32.0

* Requires 25-6FE1 Adapter

HARRIS Cont.	Device	Code	Rev
	7640	5831	32.0
	7641	5831	32.0
	7680	5832	32.0
	7681	5832	32.0
	7680P	5832	32.0
	7681P	5832	32.0
	7680RP9	5832	32.0
	7681RP9	5832	32.0
	76160	5833	32.0
	76161	5833	32.0
	76320	5834	32.0
	76321	5834	32.0
	76641	5835	32.0
	7608	583A	32.0
	7616	583B	32.0
CMOS PROMs	6641	5A3C	30.1
	6616	5A3D	30.1

HITACHI	Device	Code	Rev
EPROMs	2532	BF44	31.0
	2716	BF42	31.0
	2732	BF44	31.0
	2732A	BFC4	31.0
	2764	BF4A	27.0
	2764A	BFCA	27.0
	27C64	BFDA	27.0
	27C101.*	BFE1	31.0
	27128	BF4B	27.0
	27128A	BFCB	27.0
	27256	BF4C	27.0
	27C256	BFDC	29.0
	27C301.**	BFE3	31.0
	27512	BF4D	29.0
EEPROMs	48016	BF51	31.0
	58064	BF5A	31.0
	58C65	BF7C	31.0

HUGHES	Device	Code	Rev
EEPROMs	3004	03B8	31.0
	3104A	03B8	31.0
	3704	03B8	31.0
	3008	03B9	31.0
	3108A	03B9	31.0
	3708	03B9	31.0

ICT	Device	Code	Rev
EPROMs	27C64A	0CDA	30.0

INTEL	Device	Code	Rev
EPROMs	27010..*	6FE1	31.0
	27011	6FCE	31.0
	2716	6F42	31.0
	2732	6F44	31.0
	2732A	6FC4	31.0
	2764	6F4A	27.0
	2764A	6FCA	31.0
	27C64	6FDA	33.0
	27128	6F4B	27.0
	27128A	6FCB	31.0
	27128B	6FCB	31.0
	27C128	6FDB	27.0
	27256	6F4C	31.0
	27C256	6FDC	31.0
	27512	6F4D	31.0
	27513	6FCD	35.0
	27C513	6FCD	35.0
	87C257	6FED	31.0
Quick Pulse	2764	6FFA	32.0
	27128	6FFB	32.0
	27256	6FFC	32.0
	27512	6FFD	32.0
EEPROMs	2816A	6F52	31.0
	2816B	6F57	31.0
	2817A	6F58	31.0
	2864A	6F5C	31.0
	28256	6F5F	31.0

* Requires 25-6FE1 Adaptor

** Requires 25-BFE3 Adaptor

NATIONAL SEMICONDUCTOR	Device	Code	Rev
EPROMs	2716	3F42	31.0
	27C16	3F42	31.0
	27C16B	3FD3	31.0
	2732	3F44	31.0
	2732A	3FC4	31.0
	27C32	3F44	31.0
	27C32B	3FD5	30.0
	27C32H	3FD6	35.0
	2764	3F4A	27.0
	2764A	3FCA	27.0
	27C64	3FDA	27.0
	27128	3F4B	27.0
	27128A	3FCB	27.0
	27C128	3FDB	27.0
	27256	3F4C	27.0
	27C256	3FDC	27.0
	27C512	3FDD	27.0
EEPROMs	9716	3F51	31.0
	9817	3F58	31.0
	98C64	3F7C	31.0
BIPOLAR	74S188	3801	27.0
	74S288	3801	27.0
	74S387	3802	27.0
	74S287	3802	27.0
	74S570	3803	27.0
	74S571	3803	27.0
	74S572	3805	27.0
	74S573	3805	27.0
	87S184	3808	27.0
	87S185	3808	27.0
	74S471	3810	27.0
	74S473	3812	27.0
	74S472	3812	27.0
	87S195	3816	27.0
	87S280	3822	27.0
	87S281	3822	27.0
	27S290	3823	27.0
	27S291	3823	27.0
	74S474	3831	27.0
	74S475	3831	27.0
	87S180	3832	27.0
	87S181	3832	27.0
	87S190	3833	27.0
	87S191	3833	27.0

NEC	Device	Code	Rev
EPROMs	2716	CF42	31.0
	2732	CF44	31.0
	2732A	CFC4	31.0
	2764	CF4A	27.0
	2764A	CFCA	27.0
	27C64	CFDA	30.0
	27128	CF4B	27.0
	27128A	CFCB	27.0
	27256	CF4C	27.0
	27C256	CFDC	30.0
	27C256A	CFCC	30.0
	27C512	CFDD	35.0
	27C1000.**	CFE3	31.0
	27C1001.*	CFE1	31.0
EEPROMs	28C64	CF7C	31.0
BIPOLAR PROMs	406	C805	27.0
	426	C805	27.0
	409	C833	27.0
	429	C833	27.0

OKI	Device	Code	Rev
EPROMs	2716	0842	31.0
	2532	0843	31.0
	2732	0844	31.0
	2732A	08C4	31.0
	2764	084A	30.0
	2764A	08CA	30.0
	27128	084B	31.0
	27128A	084B	30.0
	27256	084C	30.0
	27512	084D	30.0
EEPROMs	2816A	0852	31.0

* Requires 25-6FE1 Adaptor

** Requires 25-BFE3 Adaptor

RAYTHEON	Device	Code	Rev
BIPOLAR	29660	2B02	34.0
PROMs	29663	2B02	34.0
	29610	2B03	34.0
	29613	2B03	34.0
	29650	2B08	34.0
	29653	2B08	34.0
	29600	2B10	34.0
	29603	2B10	34.0
	29620	2B12	34.0
	29623	2B12	34.0
	29640	2B1A	34.0
	29643	2B1A	34.0
	29624	2B31	34.0
	29627	2B31	34.0
	29630	2B32	34.0
	29633	2B32	34.0
	29680	2B33	34.0
	29683	2B33	34.0
	29670	2B34	34.0
	29673	2B34	34.0
	29683S	2B23	34.0
	29673S	2B24	34.0
	29634	2B3A	34.0
	29637	2B3A	34.0

SEEQ	Device	Code	Rev
EPROMs	5133	FF4A	27.0
	2764	FF4A	27.0
	5143	FF4B	27.0
	27128	FF4B	27.0
	27256	FF4C	27.0
	27C256	FFDC	27.0
EEPROMs	5516A	FF52	31.0
	2816A	FF52	31.0
	2816AH	FF55	31.0
	2817A	FF58	31.0
	2817AH	FF59	31.0
	2864	FF5B	31.0
	28C64	FF7C	31.0
	28C256	FF7F	31.0
	36C16	FFBA	31.0
	36C32	FFBB	31.0
	38C16	FF76	31.0
	38C32	FF7A	31.0
	48128	FFAE	31.0
	5213	FF54	31.0
	52B13	FF54	31.0
	52B13H	FF56	31.0
	5223	FF55	31.0
	52B33	FF5A	31.0
	52B33H	FF5E	31.0

ROCKWELL	Device	Code	Rev
EPROMs	87C32	06D4	31.0
EEPROMs	2816A	0652	31.0

SAMSUNG	Device	Code	Rev
EEPROMs	2816A	0952	31.0
	2864A	095C	31.0
	2865A	095C	31.0

SGS	Device	Code	Rev
EPROMs	2532	8F43	31.0
	2716	8F42	31.0
	2732A	8FC4	31.0
	2764	8F4A	30.0
	2764A	8FCA	30.0
	27128A	8FCB	30.0
	27256	8F4C	30.0
	27512	8F4D	30.0

SIGNETICS	Device	Code	Rev
EPROMs	27C64A	1FDA	30.0
	27C256	1FDC	30.0
BIPOLAR	82S23	1801	27.0
PROMs	82S123	1801	27.0
Generic 1	82S126	1802	27.0
	82S129	1802	27.0
	82S130	1803	27.0
	82S131	1803	27.0
	82S136	1805	27.0
	82S137	1805	27.0
	82S184	1808	27.0
	82S185	1808	27.0
	82S135	1811	27.0
	82S146	1812	27.0
	82S147	1812	27.0
	82S140	1831	27.0
	82S141	1831	27.0
	82S180	1832	27.0
	82S180A	1832	27.0
	82S180B	1832	27.0
	82S181	1832	27.0
	82S181A	1832	27.0
	82S181B	1832	27.0
	82S191	1833	27.0
	82S191A	1833	27.0
	82S321	1834	27.0
	82S115.****	1839	33.0
	82S2708	183A	27.0
BIPOLAR	82HS195	191A	27.0
PROMs	82HS321	1934	27.0
Generic 2	82HS641	1935	27.0

SMOS	Device	Code	Rev
EPROMs	27C64	OFDA	30.0
	27128	OF4B	30.0
	27C256	OFDC	30.0
EEPROMs	2864	OF5B	31.0

TEXAS INSTRUMENTS	Device	Code	Rev
EPROMS	2516	4F42	31.0
	2532	4F43	31.0
	2532A	4FC5	31.0
	2732	4F44	31.0
	2732A	4FC4	31.0
Cont..	2564	4F47	31.0
	2764	4F4A	27.0
	27128	4F4B	27.0
	27C128	4FDB	35.0
	27128A	4FCB	27.0
	27256	4F4C	27.0
	27C256	4F4C	27.0
	27C512	4FDD	30.0
SNAP	27C128	4FFB	32.0
	27256	4FFC	32.0
	27C512	4FFD	32.0
BIPOLAR	18SA030	4801	27.0
PROMs	18S030	4801	27.0
	18SA10	4802	27.0
	18S10	4802	27.0
	18SA22	4811	27.0
	18S22	4811	27.0
	18SA42	4812	27.0
	18S42	4812	27.0
	18SA46	4831	27.0
	18S46	4831	27.0
	24S10	4902	27.0
	24SA10	4902	27.0
	24S41	4906	27.0
	24SA41	4906	27.0
	24S81	4908	27.0
	24SA81	4908	27.0
	28L22	4911	27.0
	28LA22	4911	27.0
	28S42	4912	27.0
	28L42	4912	27.0
	28P42	4912	27.0
	28S45	4931	27.0
	28L45	4931	27.0
	28P45	4931	27.0
	28S85	4932	27.0
	28SA85	4932	27.0
	28L85	4932	27.0
	28P85	4932	27.0
	28P86	4932	27.0
	28S166	4933	27.0
	28SA166	4933	27.0
	28L166	4933	27.0
	28P166	4933	27.0
	28S2708	493A	27.0
	34S10	4A02	31.0
	34SA10	4A02	31.0

**** Requires 20-1839 Adaptor

TEXAS INSTRUMENTS	Device	Code	Rev
Cont.....	34S162.....	4A1A...	31.0
	34SA162.....	4A1A...	31.0
	34S41.....	4A05...	31.0
	34SA41.....	4A05...	31.0
	34SR6165.....	4A2E...	31.0
	34SRA165.....	4A2E...	31.0
	38S030.....	4A01...	31.0
	38SA030.....	4A01...	31.0
	38S22.....	4A11...	31.0
	38SA22.....	4A11...	31.0
	38S42.....	4A12...	31.0
	38SA42.....	4A12...	31.0
	38R85.....	4A28...	31.0
	38RA85.....	4A28...	31.0

THOMSON	Device	Code	Rev
Bipolar.....	71190.....	1A33...	27.0
Proms	71191.....	1A33...	27.0
	71640.....	1A35...	30.1
	71641.....	1A35...	30.1

TOSHIBA	Device	Code	Rev
EPROMs.....	2716.....	EF42...	31.0
	2732.....	EF44...	31.0
	2732A.....	EFC4...	31.0
	2764.....	EF4A...	27.0
	2764A.....	EFCA...	31.0
	27128.....	EF4B...	27.0
	27128A.....	EFCB...	31.0
	27256.....	EFDC...	34.0
	27256A.....	EFCC...	31.0
	27512.....	EF4D...	31.0
	57256.....	EFDC...	34.0
	57256A.....	EFD8...	31.0
	571000...*...	EFE1...	31.0
	571001...**..	EFE3...	31.0
High Speed...	2764A.....	EFFA...	32.0
Programming	27128A.....	EFFB...	32.0
II	27256A.....	EFFC...	32.0
	27512.....	EFFD...	32.0
	57256A.....	EFF8...	32.0

3-RAIL	Device	Code	Rev
EPROMs.....	2716.....	EC3F...	30.1
	2704.....	EC40...	30.1
	2708.....	EC41...	30.1

VITELIC	Device	Code	Rev
EPROMs.....	27C256.....	ODDC...	30.0

VLSI	Device	Code	Rev
EPROMs.....	27C64.....	04DA...	30.0
	27C128.....	04DB...	30.0
	27C256.....	04DC...	30.0

WAFERSCALE	Device	Code	Rev
EPROMs.....	57C64.....	OBDA...	30.0
	57C128.....	OBDB...	30.0
	57C191.....	OBAA...	34.0
	57C291.....	OBA9...	34.0

XICOR	Device	Code	Rev
EEPROMs.....	2804A.....	0750...	31.0
	2816A.....	0752...	31.0
	2816B.....	0757...	31.0
	2816H.....	0757...	31.0
	28C16.....	0774...	31.0
	2864A.....	075C...	31.0
	2864B.....	075D...	31.0
	2864H.....	075D...	31.0
	28C64.....	077C...	31.0
	28256.....	075F...	31.0
	28C256.....	077F...	31.0

* Requires 25-6FE1 Adaptor

** Requires 25-BFE3 Adaptor

Operating Manual

for

PPZ Universal programmer with Zm2000 Universal Z-Module

Manual Revision 2

Stag Electronic Designs Ltd.
Stag House
Tewin Court
Tewin Road
Welwyn Garden City
Herts
AL7 1AU
Telephone (0707) 332148
Telex 8953451

(Rev. 2) Address-01

803 1000

Address-02

Addendum

Note: These additions will be incorporated into the Zm2000 manual at the next revisional reprint.

Additions to Existing Manual (PPZ/Zm2000 Rev.2)

New RAM Editor Features

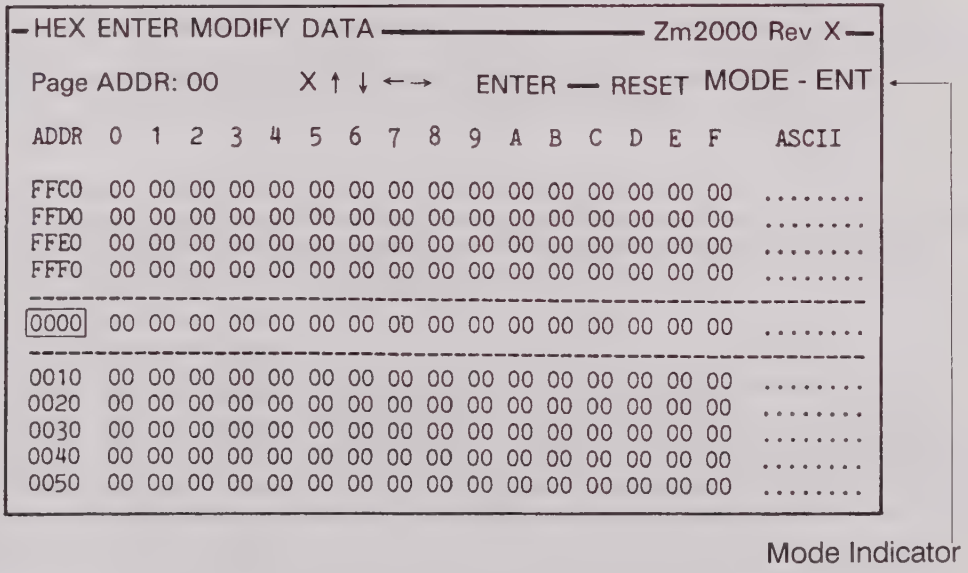
Description

The RAM Editor in the Zm2000 module now allows the insertion and deletion of bytes of data as well as the usual 'overwrite' mode.

Operation

Enter the required RAM editor as usual.(See Section 5.2 of the manual).It should be noticed that there is an additional message in the top right hand corner of the screen. This is the Mode Indicator.

The editor can be used in the usual way - overwrite mode(**ENT**) - or it can be used in **DEL**ete or **INS**ert modes.



Mode Indicator

Insert Mode

With the cursor still in the default position on the address column, press Key L. This will change the mode from 'overwrite' to Insert. The mode indicator will change from ENT to INS.

Select the location at which a data byte is to be inserted using the cursor keys only. If the data address is very high, it may be quicker to pre-select it before entering the Insert mode.

CONTINUED

When the cursor is positioned on the required byte of data, press Enter. All data at and above the selected address will be repositioned one location higher within the RAM. A new data byte can now be keyed in.

Please note: The last data byte in the available RAM will be lost.

Delete Mode

With the cursor still in the default position on the address column, press Key L twice. This will change the mode from 'overwrite' to Delete. The mode indicator will change from ENT to DEL.

Select the location at which a data byte is to be deleted using the cursor keys only. If the data address is very high, it may be quicker to pre-select it before entering the Delete mode. When the cursor is positioned on the required byte of data, press Enter. All data above the selected address will be repositioned one location lower within the RAM. The last available byte in the RAM will become 00.

Device Selection

The device selection sequence is now slightly different to that detailed in the manual.

Press Key 1

This will display the available device types and the list of manufacturers available in the chosen device category.

If known, a 4 digit manufacturer and pinout code can be entered followed by Enter and Reset. If the full code is unknown, then the manufacturer can be selected with the cursor keys followed by Enter. This will then display a list of the devices available for that manufacturer. The device can be chosen with the cursor keys or by entering the relevant 2 digit pinout code. Press Reset.

List Device

The List Device feature can now handle 'paged' devices. To select the required 'page' from within the List Device feature, Press Key H and key in the page number. Press Enter. The new page will be displayed.

Light Pen 'On/Off' Option

This option is only available with the Light Pen Pack, Zp006 installed.

On power-up, the light pen will be 'On'. To turn the light pen off, select Set System (Press pen on 'A...Set System') then select Miscellaneous (Press pen on '4...Miscellaneous'). The light pen On/Off option is displayed at the bottom. Press the pen on 'Off' followed by Reset. The light pen will now be turned off and the keyboard will be enabled.

To turn the light pen back on, select Set System (Key A) and then Miscellaneous (Key 4). Press the down cursor key 3 times. Light Pen 'Off' will be shown flashing. Press the right cursor key once followed by Reset. The light pen will be turned on and the keyboard disabled.

CONTINUED

Changing the Number of 'Nulls' for Data Output.

Prior to Software Revision 33, the number of nulls following a CR,LF was set to 4. From Software Revision 33 onwards, the number of nulls could be set, by the user, from 0 to 72.

To do this, from the System Menu:

Press Key A followed by Key 2.

This will display the Set I/O Software screen.

The 'Number of Nulls' option can be selected using the vertical cursor keys and the value can be entered from the hex. keyboard. Only values from 0 to 72 can be used. If a larger value is entered, the display will reset to 72 when Enter is pressed.

Having entered the required value, press Enter to confirm the selection followed by Reset to return the System Menu.

Zm2000 MODULE

CONTENTS

Section 1 GENERAL INTRODUCTION
- 1.1 PPZ Mainframe
- 1.2 Keyboard
- 1.3 Selecting the Operating Mode
- 1.4 Initial Setting-up Procedure
- 1.5 Horn Control
- 1.6 Z-Modules
- 1.7 Module Setting-up Procedure
- 1.8 RAM Operating Transfers

Section 2. DEVICE SELECTION
- 2.1 Device Selection
- 2.2 Device Description
- 2.3 Electronic Identifier

Section 3. ADDRESS INCREMENT
- 3.1 Address Increment
- 3.2 8-Bit Mode (Address Increment . . . 1)
- 3.3 16-Bit Mode (Address Increment . . . 2)
- 3.4 32-Bit Mode (Address Increment . . . 4)
- 3.5 Nibble Swap (Increment on 4-Bit Devices)

Section 4. DEVICE FUNCTIONS
- 4.1 Load
- 4.2 Test Operations
- 4.3 Empty Test (Manual)
- 4.4 Pre-Program Checks
- 4.5 Programming
- 4.6 Verify Test (Manual)
- 4.7 Checksum
- 4.8 Cyclic Redundancy Check (CRC)
- 4.9 List Device

Section 5. RAM FUNCTIONS
- 5.1 List RAM
- 5.2 Edit Ram
- 5.2.1 Hex Editor
- 5.2.2 Decimal Editor
- 5.2.3 Octal Editor
- 5.2.4 Logical 'OR' Data
- 5.2.5 Logical 'AND' Data
- 5.2.6 Nibble Swap
- 5.2.7 String Search
- 5.2.8 Block Move
- 5.2.9 Complement
- 5.3 Fill RAM
- 5.3.1 Fill RAM with 00s
- 5.3.2 Fill RAM with FFs
- 5.3.3 Fill RAM with 'EMPTY' State
- 5.3.4 Fill RAM Arbitrarily

Section 6 INTERFACE
6.1 Setting the I/0 Interface Parameters
6.2 I/O Mode
6.2.1 Input
6.2.2 Output

Section 7 FORMAT DESCRIPTIONS
7.1 Interface Formats
7.2 TEK HEX
7.3 Extended TEK HEX
7.4 INTELLEC 86
7.5 MOTOROLA S-Record
7.6 HEX ASCII
7.7 STAG HEX*
7.8 Binary and DEC Binary
7.9 MOS Technology

Section 8 RS232C HARDWARE DESCRIPTIONS
8.1 RS232C Interface Port Connections
8.2 Connection types
8.3 Hardware Handshake

Section 9 REMOTE CONTROL
9.1 Selecting the Operating Modes
9.2 Setting-up procedure for Remote Control (Serial Interfaces)
9.3 Remote Control Commands
9.4 Setting-up procedure for Remote Control (IEEE Interface)

Section 10. PASS-THROUGH
10.1 Pass-Through

Section 11. Light Pen Operating Instructions

Section 12 Handler Pack Operating Instructions

SECTION 1

stag

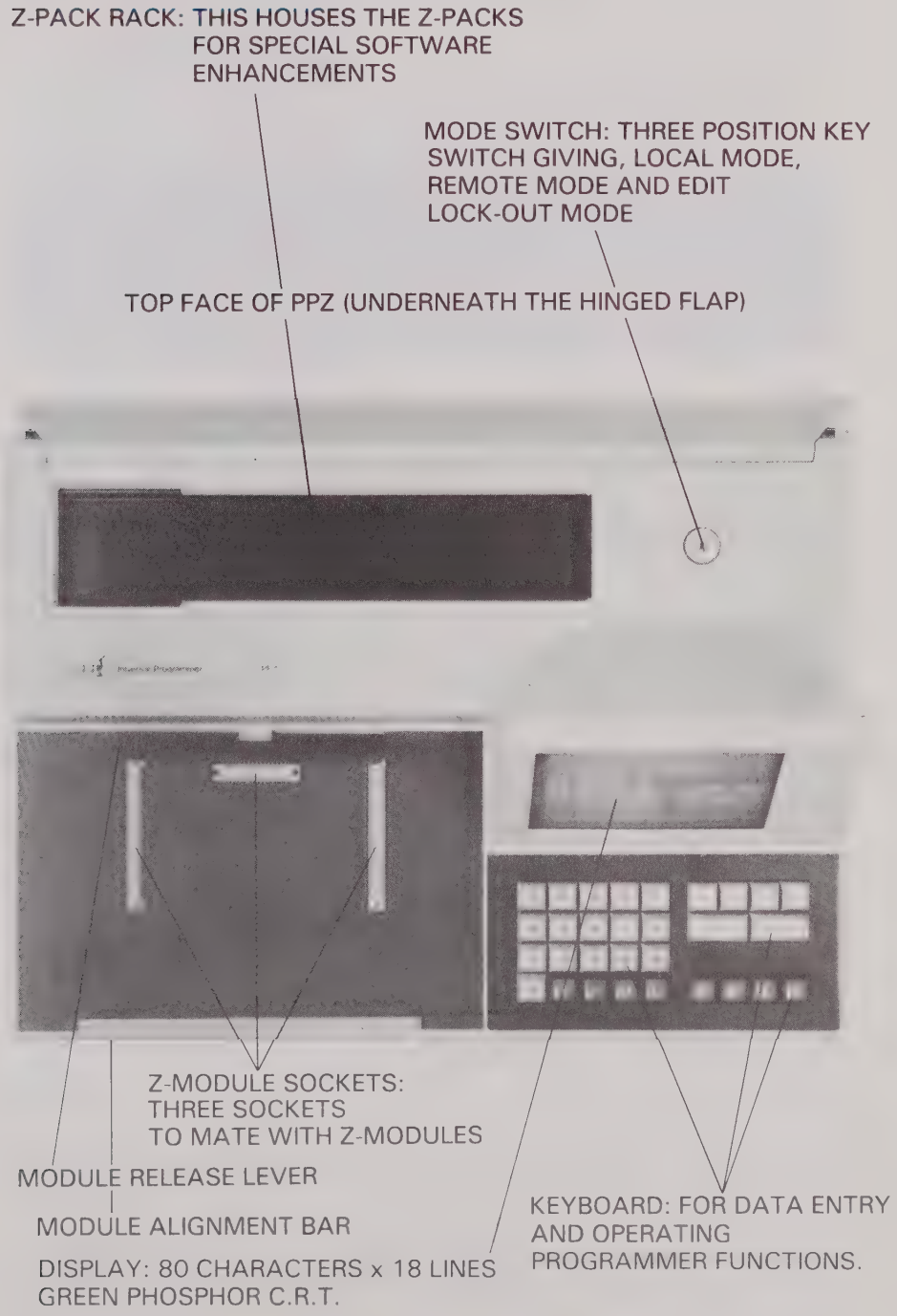
Sophisticated systems for the discerning engineer.

1.—GENERAL INTRODUCTION

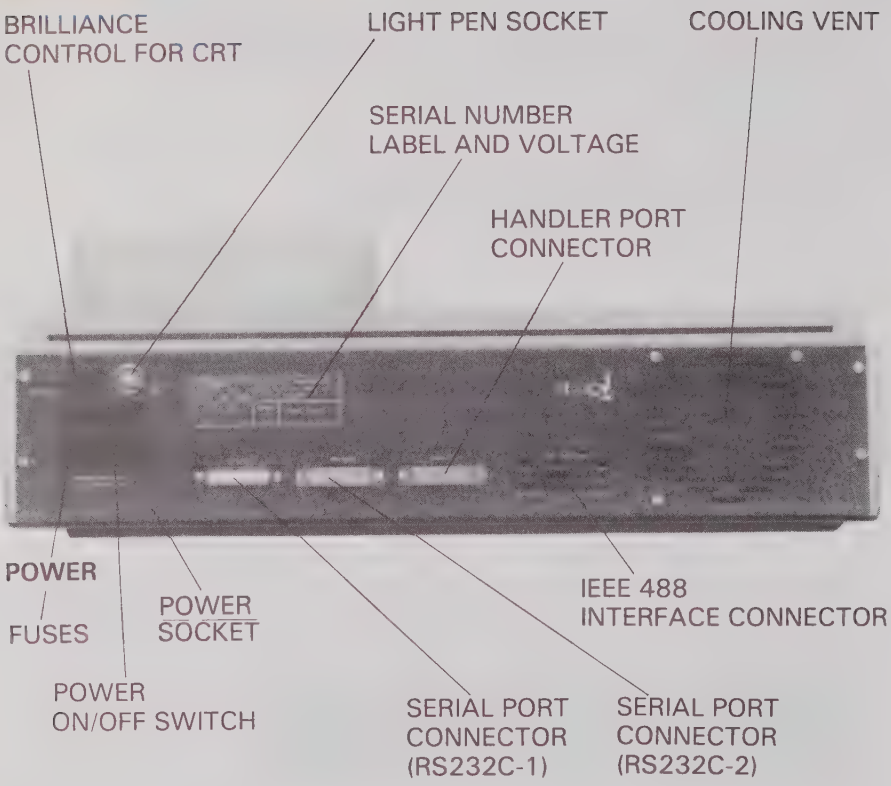
INTRODUCTION

The PPZ is a Universal programmer designed with the intention of programming any programmable part in any technology.

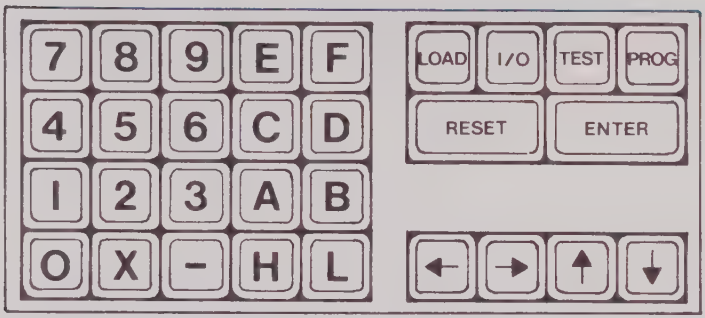
1.1 PPZ MAINFRAME

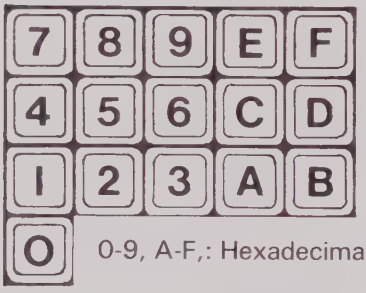

Z-PACK RACK: THIS HOUSES THE Z-PACKS FOR SPECIAL SOFTWARE ENHANCEMENTS
MODE SWITCH: THREE POSITION KEY SWITCH GIVING, LOCAL MODE, REMOTE MODE AND EDIT LOCK-OUT MODE
TOP FACE OF PPZ (UNDERNEATH THE HINGED FLAP)
Z-MODULE SOCKETS: THREE SOCKETS TO MATE WITH Z-MODULES
MODULE RELEASE LEVER
MODULE ALIGNMENT BAR
DISPLAY: 80 CHARACTERS x 18 LINES GREEN PHOSPHOR C.R.T.
KEYBOARD: FOR DATA ENTRY AND OPERATING PROGRAMMER FUNCTIONS.

REAR PANEL


BRILLIANCE CONTROL FOR CRT
LIGHT PEN SOCKET
COOLING VENT
SERIAL NUMBER LABEL AND VOLTAGE
HANDLER PORT CONNECTOR
POWER
FUSES
POWER SOCKET
POWER ON/OFF SWITCH
SERIAL PORT CONNECTOR (RS232C-1)
SERIAL PORT CONNECTOR (RS232C-2)
IEEE 488 INTERFACE CONNECTOR

1.2 KEYBOARD—for data entry and operating programmer functions





0-9, A-F,: Hexadecimal Keyboard.

XHL: Entry keys for PAL* and IFL*

Four cursor control keys

To load the programmer RAM

I/O Key—to execute input or output via the I/O ports

TEST Key—to initiate a test function (empty, verify etc.)

PROGRAM Key—to initiate a sequence of tests and the program routine.

RESET Key—to return the programmer to its initial state.

ENTER Key—to enter a selected parameter or to execute a pre-selected function.

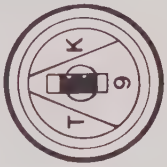
1.3 SELECTING THE OPERATING MODES

Selection of the operating mode is made by use of the three position key-switch located underneath the hinged flap. By inserting the supplied key one of these modes can be chosen.

LOCAL MODE (arrow vertical)

In LOCAL mode all functions of the PPZ are controlled from its own keyboard. All the functions operate interactively with the CRT and Input/Output transmissions take place between the PPZ and the peripheral equipment under 'Local Control'.

REMOTE MODE (arrow pointing to the left)



In REMOTE the PPZ operates under remote control from a computer or a terminal. The keyboard of the PPZ is inoperative at this time and the CRT will only display information as requested under remote command.

EDIT LOCK-OUT MODE (arrow pointing to the right)

The Edit Lock-out mode ensures all editing key functions are inoperative but that all other functions operate normally under LOCAL MODE conditions.

In this mode data corruption or changes to the RAM data cannot take place and security of information is ensured.

Interchange between LOCAL and REMOTE MODE can be accomplished by turning the key-switch to the required direction and pressing RESET.

Interchange between LOCAL and EDIT LOCK-OUT MODE can be made by simply turning the key to the required direction. (Should an edit key function be requested under Edit Lock-Out Mode the CRT will display: EDIT LOCKED OUT)

1.4 INITIAL SETTING-UP PROCEDURE

Before attempting to power-up your PPZ Programmer ensure that it is set to the correct operating voltage in line with your power supply. The voltage setting is printed on the rear panel.

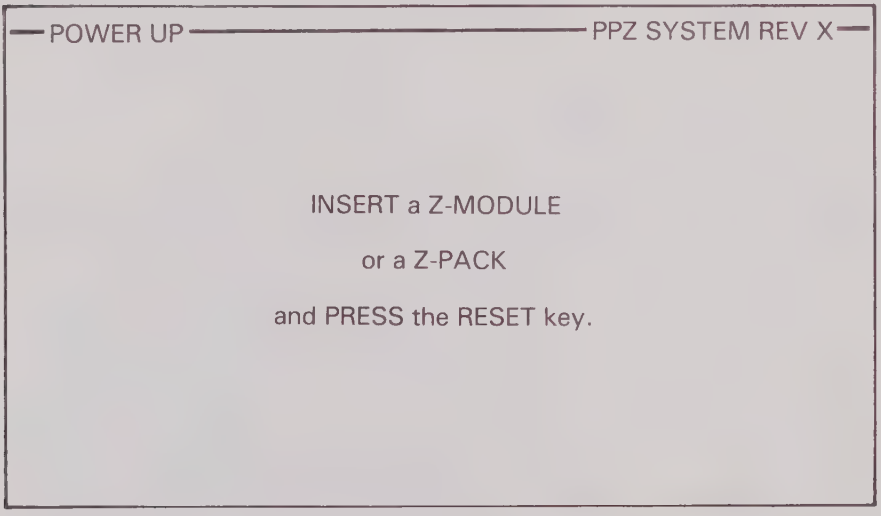
1. Plug the supplied power-cord into the rear panel socket.

2. Connect power-cord to power supply.

3. Power-up the PPZ using the ON/OFF switch on the rear panel.

4. Wait a few seconds for the CRT to warm up and then adjust the picture brightness using the brilliance control on the rear panel.

After ''POWER-UP'' and without a Z-Module inserted, the display will read:

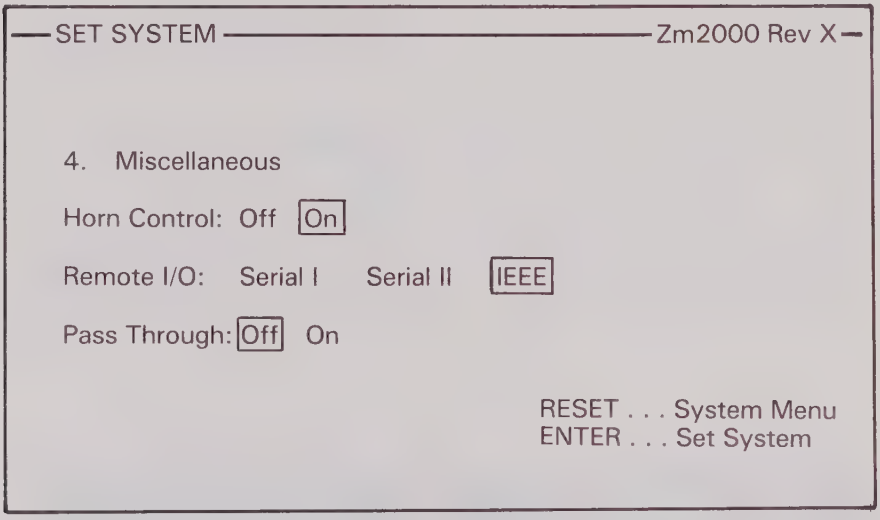


The CRT has a display protection mechanism to save the screen from burning. Therefore, the display will shut-off automatically if the screen is not in use for a period in excess of 4 minutes. (To restore the picture, press any of the cursor keys.)

1.5 HORN CONTROL

To determine whether the module should operate with the warning horn activated or silent:

Press KEY A followed by KEY 4, to display:


SET SYSTEM
Zm2000 Rev X
4. Miscellaneous
Horn Control: Off [On]
Remote I/O: Serial I Serial II [IEEE]
Pass Through: [Off] On
RESET . . . System Menu
ENTER . . . Set System

The arrow keys can be used to move the cursor to select 'off' or 'on'.

Press ENTER

to confirm selection

The Set System Menu is re-displayed

Press RESET

to return to the System Menu.

1.6. Z-Modules

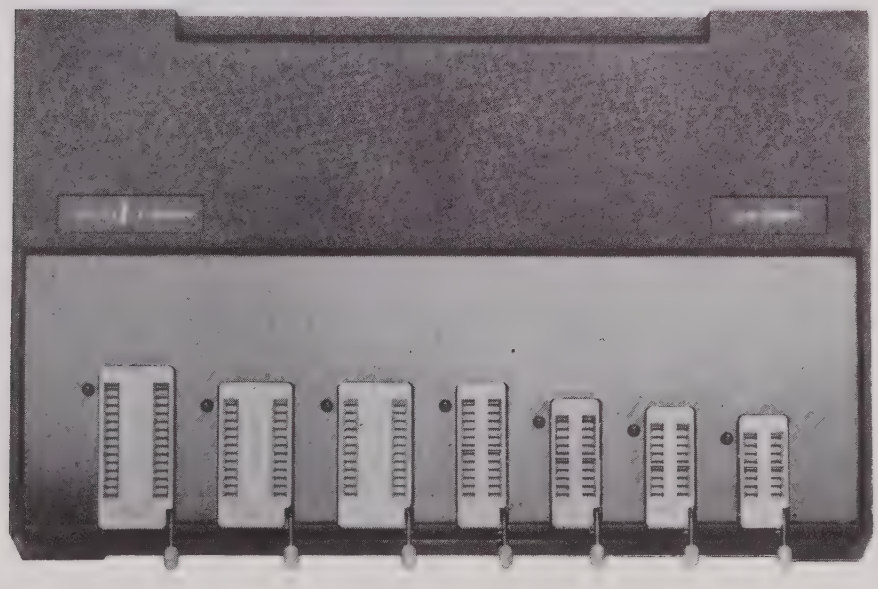
The PPZ is a Universal Programmer designed with the intention of programming any programmable part in any technology.

Programmable devices are divided into groups and a range of Z-modules are available which plug into the PPZ to support the programming of these different families. The current range of Z-modules are as follows:

Zm2000 Universal PROM Module

This Z-Module supports NMOS, HMOS, CMOS EPROMs and EEPROMs, Fusible Link, AIM and DEAP bipolar PROMs and some CMOS PROMs.

The module features the latest 'intelligent algorithms' to speed programming times of EPROMs and it will interpret 'electronic identifier' coded devices which automatically selects the correct algorithm for that device. The multisocket approach is optimised to give high programming speeds and this combined with Stag's unique design, guarantees higher yields.



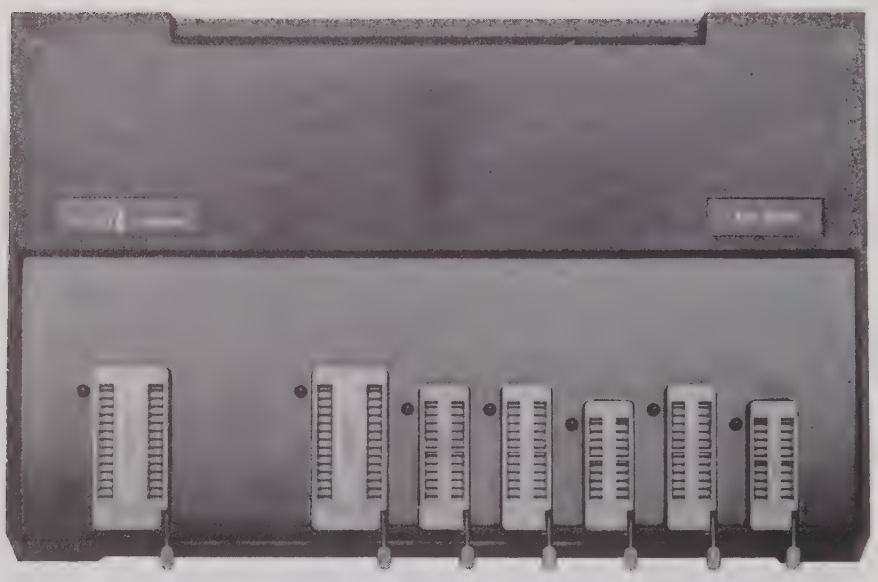
Zm2200 Logic Module

This module will program logic devices in 20, 24 and 28 pin DIL packages from Altera, AMD, Cypress, Harris, MMI, National, Signetics, Texas Instruments and VTI in one unit. No additional adaptors are required to support different manufacturers.

The Module contains operating software for both PAL* and IFL* devices and enables truth table entry and editing of data using **X-HLA**·

A special socket on the module provides a storage facility for Boolean equations and test vectors. Devices can then be tested and Boolean source information recalled for further editing at any time.

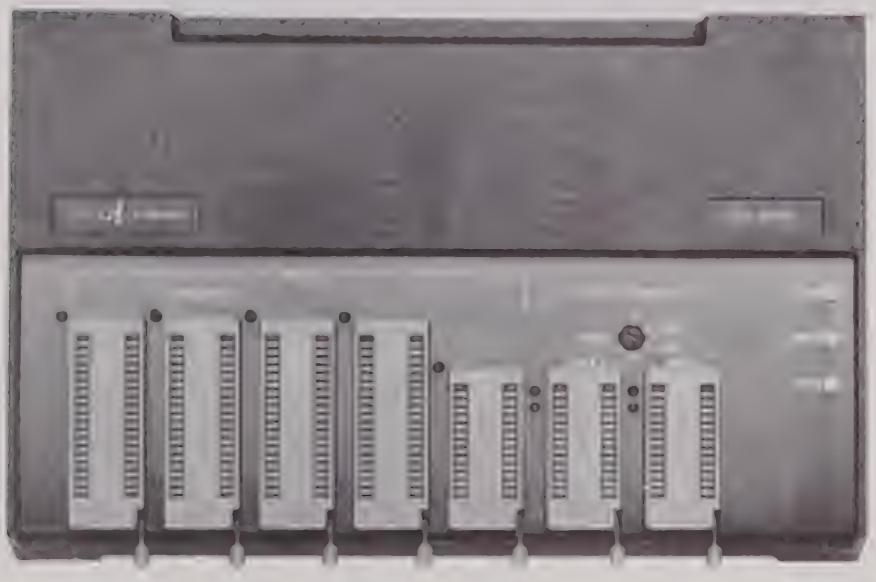
The module software is JEDEC format compatible. It will therefore accept a wide range of remotely generated data using this format. PALASM*, Signetics and other I/O formats are also supported.



Zm2500 MOS Module

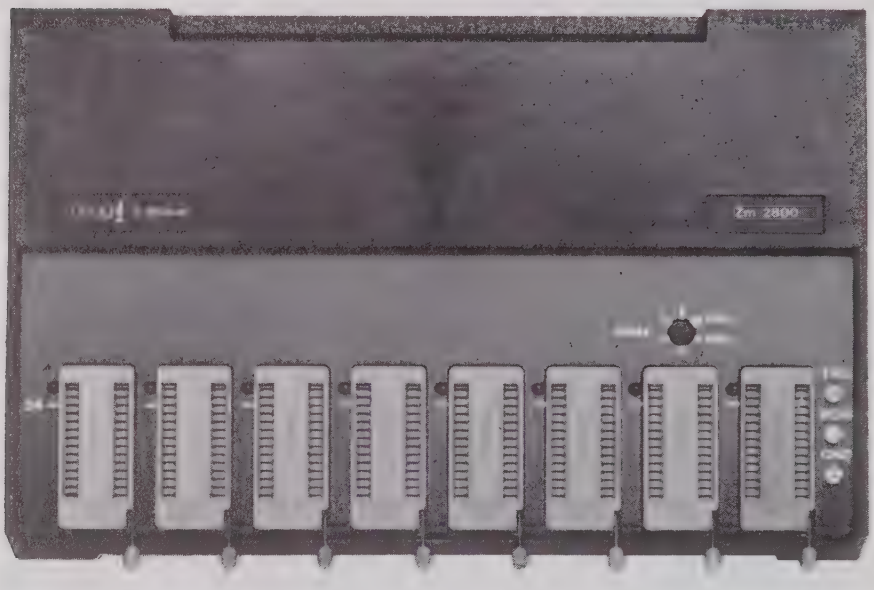
This module will program EPROMs and EEPROMs. In addition this module will program a range of programmable microprocessor devices.

A set of two EPROMs can be down loaded for 16 bit systems and programmed simultaneously as a set, or two EPROMs can be programmed as a mini-ganger.



Zm2800 Gang-Setter* for EPROMs

This module programs a set of up to eight 2764 EPROMs each with different patterns simultaneously or a set of 27256 EPROMs with memory extension. The module enables any combination of sets to be programmed, such as four sets of two for 16 bit word applications, all in one operation. This allows rapid programming of sets of PROMs for prototyping or for limited production runs.



Zm2300 Super PAL* Module

The most sophisticated in the range—it provides programming capability for Super PAL devices and extensive software is provided for remote editing.

Zm2900-xx Custom Module

Custom-designed for on-board programming of EPROMs.

1.7 Module Setting-up Procedure

1. Power down the PPZ with the on/off switch on the rear panel.

2. Plug in your Zm2000 module making sure the sockets on the module and the mainframe are properly aligned using the guide bar located along the front of the PPZ for lateral positioning. Make sure the module is as far forward as possible and resting against the alignment bar before pressing the module down to engage the connectors. The lever at the back of the module is only used to release the module and will automatically find its rest position when the Z-Module is properly engaged.

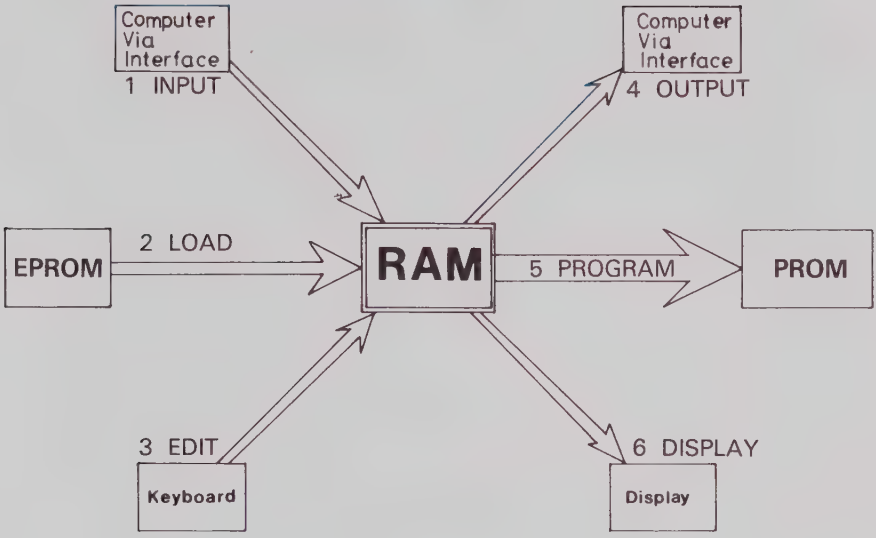
3. Check that the three position key switch under the hinged flap is indicating the required mode. (In this example the LOCAL MODE, arrow vertical).

4. Power up using the on/off switch on the rear panel.

The display will show the system menu, the module software revision number and the device last used:

```
SYSTEM MENU ──────────────────── Zm2000 Rev X

Function Codes

  1 . . . Set Device              Device: AMD Interactive
  2 . . . Device Description                2764

  3 . . . Edit RAM                LOAD  . .  Load Device
  4 . . . Fill RAM                I/O   . . .  Input/Output
                                  TEST  . . .  Test Device
  5 . . . List RAM                PROG  . . .  Program Device
  6 . . . List Device

  8 . . . Checksum
  9 . . . CRC

  A . . . Set System              Select Function: ▮
```

1.8 RAM OPERATING TRANSFERS



There are six RAM data transfer functions on the PPZ.

Data can be loaded into the RAM of the PPZ from three different sources, these are:

1. **INPUT:** From a computer, terminal or microprocessor development system through the IEEE-488 interface port or either of the RS232C interface ports.

2. **LOAD:** From a master device or devices, inserted into one of the ZIF sockets.

3. **EDIT:** Controlled from the PPZ keyboard. Function include Enter/Modify Data, Nibble Swap, Block Move, Complement, ANDing, ORing and filling the RAM with O's, F's, Arbitrarily and the Unprogrammed State.

   Data can be transfered from RAM in three ways:

4. **OUTPUT:** Through the IEEE-488 interface port or either of the RS232C interface ports to a computer, terminal or microprocessor development system.

5. **PROGRAM:** The title function whereby data is finally copied from the RAM to a device in a ZIF socket.

6. **DISPLAY:** Visual confirmation of operations undertaken.

SECTION 2

stag

Sophisticated systems for the discerning engineer.

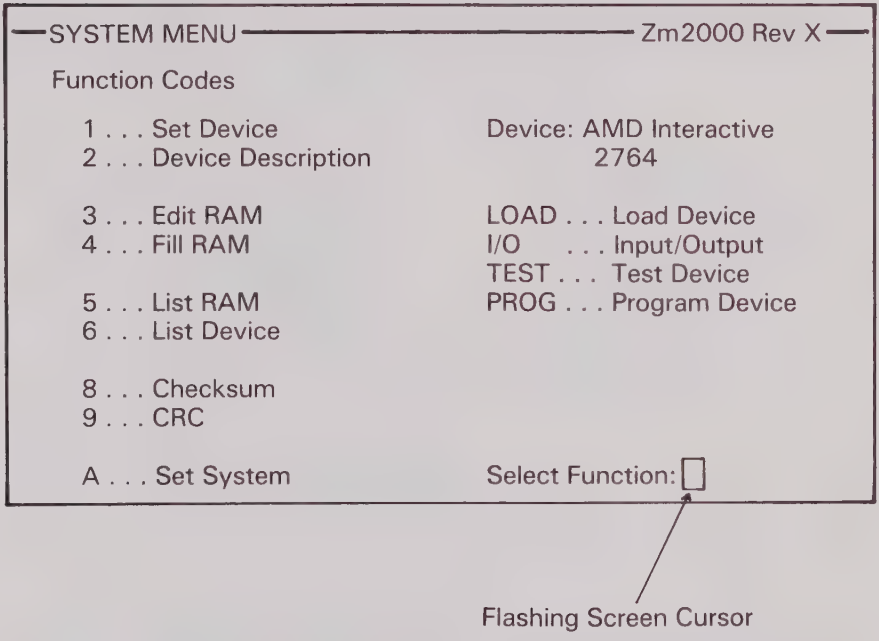
2.—DEVICE SELECTION

2.1 SELECTING THE DEVICE TO BE PROGRAMMED

A complete list of parts supported by the Zm2000 is stored within the module.

Every programmable part carries a four digit code. The first two digits define the manufacturer of the device and the second two digits represent the pinout code. (See device code list Section II).

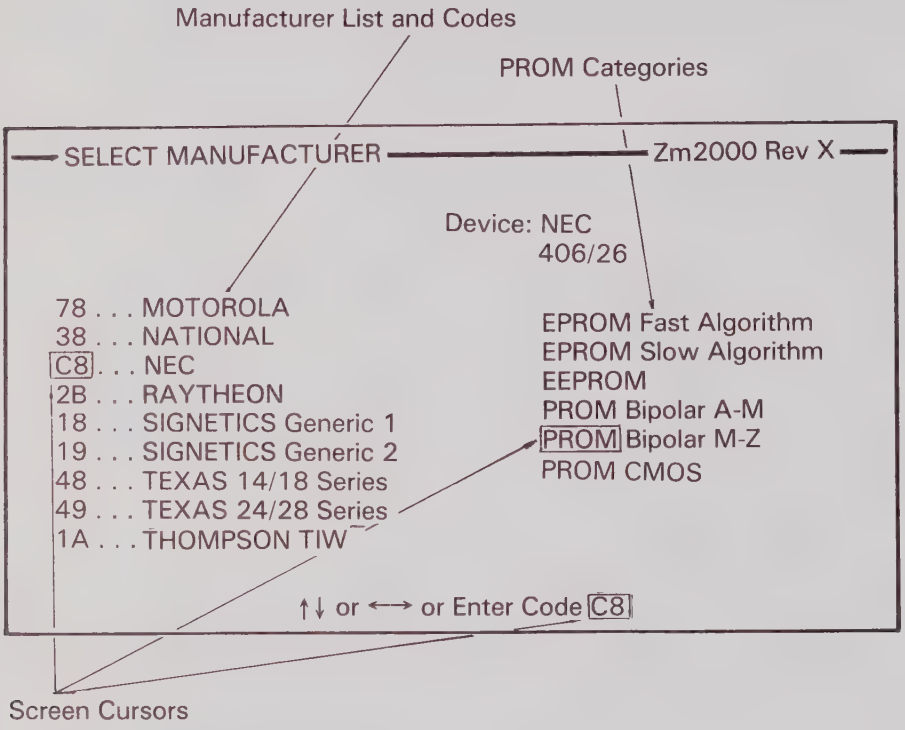
Selection of devices is made by a combination of cursor keys and device codes.


SYSTEM MENU
Zm2000 Rev X
Function Codes
1 . . . Set Device
2 . . . Device Description
3 . . . Edit RAM
4 . . . Fill RAM
5 . . . List RAM
6 . . . List Device
8 . . . Checksum
9 . . . CRC
A . . . Set System
Device: AMD Interactive
2764
LOAD . . . Load Device
I/O . . . Input/Output
TEST . . . Test Device
PROG . . . Program Device
Select Function:
Flashing Screen Cursor

As an example the following describes the selection of an AMD27512.

From the system menu, to display the selection of manufacturers available for a particular PROM type:

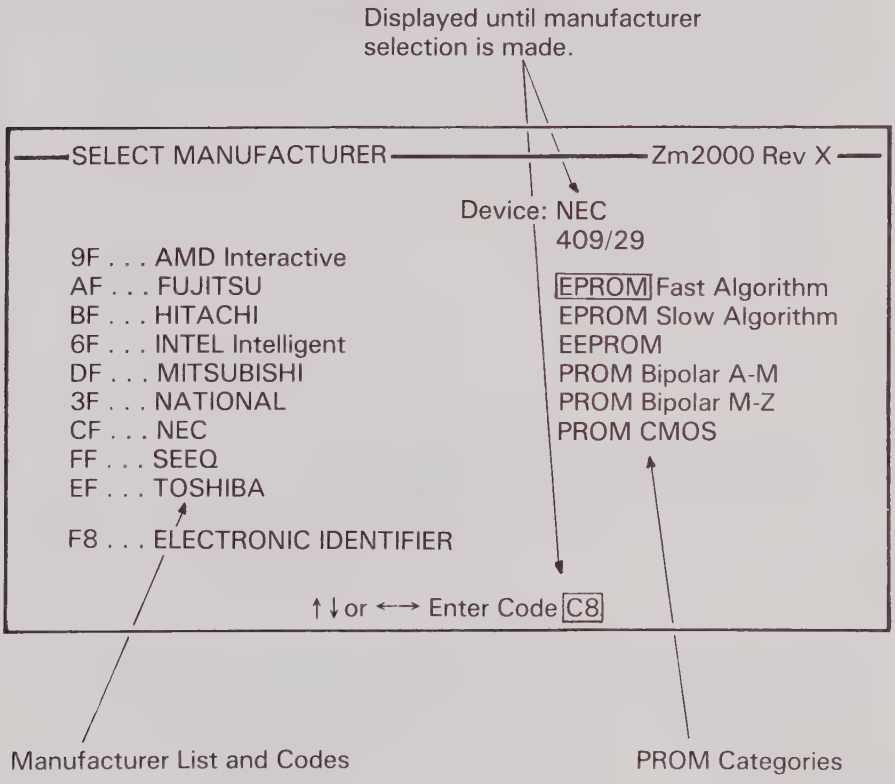
PRESS KEY 1


Manufacturer List and Codes
PROM Categories
SELECT MANUFACTURER
Zm2000 Rev X
Device: NEC
406/26
78 . . . MOTOROLA
38 . . . NATIONAL
C8 . . . NEC
2B . . . RAYTHEON
18 . . . SIGNETICS Generic 1
19 . . . SIGNETICS Generic 2
48 . . . TEXAS 14/18 Series
49 . . . TEXAS 24/28 Series
1A . . . THOMPSON TIW
EPROM Fast Algorithm
EPROM Slow Algorithm
EEPROM
PROM Bipolar A-M
PROM Bipolar M-Z
PROM CMOS
↑↓ or ←→ or Enter Code C8
Screen Cursors

The last used selection will be retained and will be indicated by screen cursors (reverse video). In the above case an NEC Bipolar PROM is indicated.

Selection of a new device must first be established through the 'PROM Category' list on the right of the screen. Interchange between manufacturer and PROM lists is made by use of the left and right cursor keys (← and →).

The Up and Down cursor keys (↑ and ↓) will locate the desired PROM type, in this case 'EPROM Fast Algorithm'.


Displayed until manufacturer selection is made.
SELECT MANUFACTURER
Zm2000 Rev X
Device: NEC 409/29
9F . . . AMD Interactive
AF . . . FUJITSU
BF . . . HITACHI
6F . . . INTEL Intelligent
DF . . . MITSUBISHI
3F . . . NATIONAL
CF . . . NEC
FF . . . SEEQ
EF . . . TOSHIBA
F8 . . . ELECTRONIC IDENTIFIER
EPROM Fast Algorithm
EPROM Slow Algorithm
EEPROM
PROM Bipolar A-M
PROM Bipolar M-Z
PROM CMOS
↑↓or ←→ Enter Code C8
Manufacturer List and Codes
PROM Categories

Manufacturers of Fast Algorithm EPROMs are now ready for selection prompted by the screen cursor.

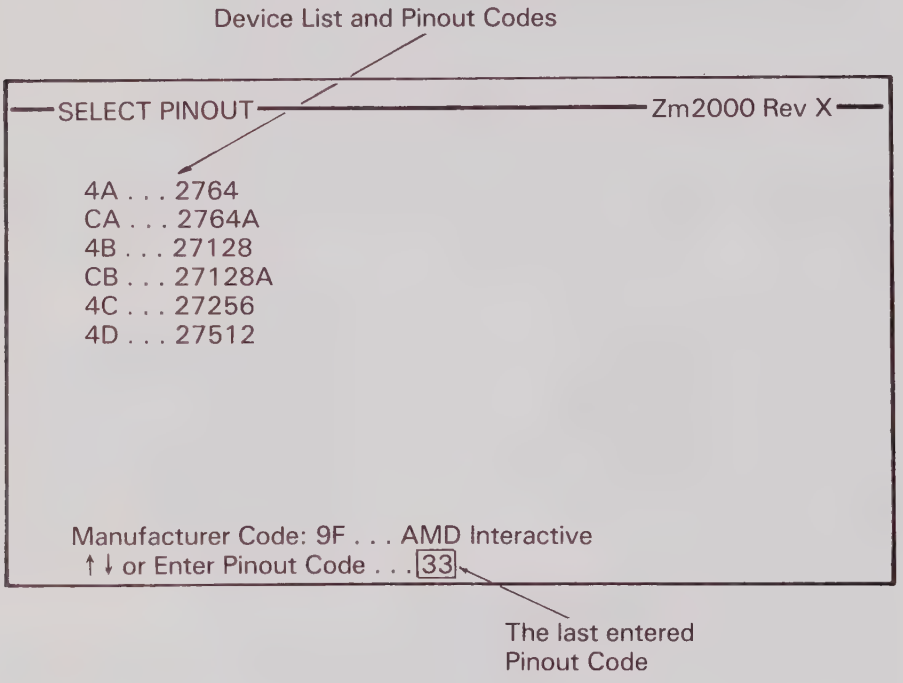
"AMD interactive" can be selected either by direct entry using the two digit manufacturers code, in this case '9F', or by use of the cursor keys. The screen cursor will indicate selection:

9F. . . AMD Interactive

When manufacturer selection is complete:

PRESS . . . ENTER

The device types available for AMD Interactive will now be displayed.


Device List and Pinout Codes
SELECT PINOUT
Zm2000 Rev X
4A . . . 2764
CA . . . 2764A
4B . . . 27128
CB . . . 27128A
4C . . . 27256
4D . . . 27512
Manufacturer Code: 9F . . . AMD Interactive
↑ ↓ or Enter Pinout Code . . . 33
The last entered
Pinout Code

Selection can again be made by direct entry using the two digit Pinout Code, in this case "4D", or by use of the Up/Down cursor keys to give:

4D . . . 27512

Full title: "AMD Interactive 27512 EPROM with fast Algorithm" Code (9F4D).

At this time a red LED will be illuminated on the Zm2000 Module adjacent to pin 1 on the ZIF socket to be used to hold the device.

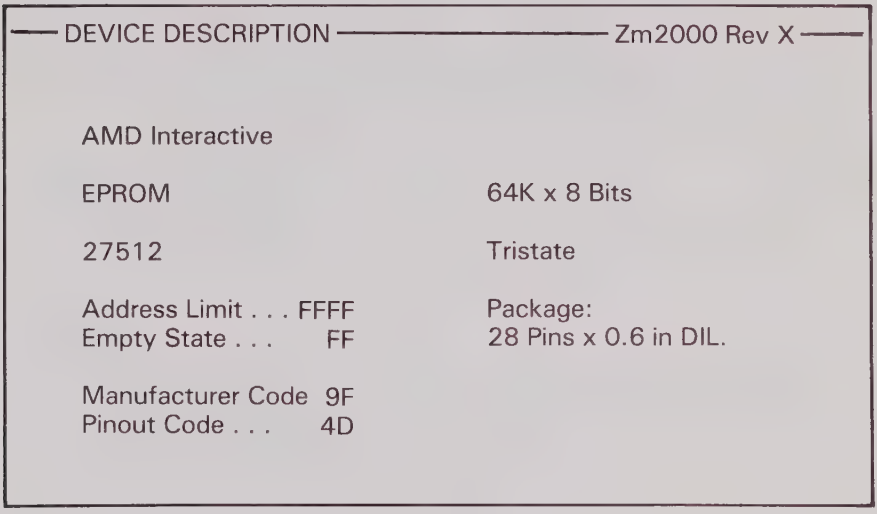
To return the system menu:

PRESS RESET

2.2 DEVICE DESCRIPTION

To display a description of the selected device from the SYSTEM MENU:

PRESS KEY 2



To return to the SYSTEM MENU:

PRESS RESET

2.3 Electronic Identifier

Electronic Identifier is a term used to describe a code mask or signature programmed into a PROM which identifies the device type and manufacturer. The code is stored outside the normal memory array and is accessed by applying 12 volts to address line A9. This allows direct identification of any device containing an Electronic Identifier and thus eliminates the need for the user to select the device type.

Important Note: Devices which do not contain an Electronic Identifier may be irreparably damaged if the Electronic Identifier function is used

TO USE THE ELECTRONIC IDENTIFIER

From the SYSTEM MENU, to enter the ELECTRONIC IDENTIFIER function,

PRESS KEY 1

This displays the manufacturer selection screen. Using the cursor keys, select the 'EPROM Fast Algorithm' category. (This is described in the section on Device Selection).

To engage the Electronic Identifier,

PRESS ← then

PRESS ↓ until screen cursor reaches the bottom of the column.

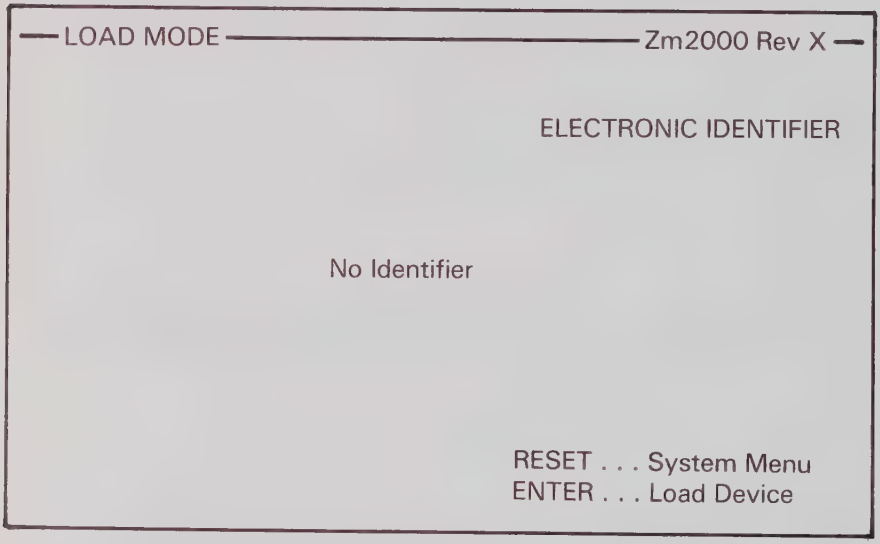
This should show the screen cursor flashing on the 'F8 . . . Electronic Identifier' line. To confirm the selection of Electronic Identifier,

PRESS ENTER

This will return the SYSTEM MENU.

Using the Electronic Identifier

If the device function keys of LOAD, PROGRAM or TEST are pressed, the PPZ will attempt to read the signature of the device in the relevant ZIF socket. If a signature is recognised, the identity of the device will be displayed on the screen. The chosen function can then be implemented. If a signature is not found, then the screen will display the message ''No Identifier'', as shown below.



To return to the SYSTEM MENU:

PRESS RESET

Should more than one type of device need to be LOADED or PROGRAMMED from the same RAM data, then the PPZ will attempt to identify each successively inserted device upon initiation of the function.

SECTION 3

stag

Sophisticated systems for the discerning engineer.

3. — ADDRESS INCREMENT

3.1 ADDRESS INCREMENT OF 8-BIT DEVICES

This is a feature related to the device functions of:

PROGRAM
LOAD
TEST (VERIFY/EMPTY)

and the RAM functions of:

FILL RAM
LOGICAL 'OR' DATA
LOGICAL 'AND' DATA
NIBBLE SWAP
COMPLEMENT
CYCLIC REDUNDANCY CHECK

Increment is an increase in the amount of data bytes that constitute the size of a word within the RAM (1 byte = 8 bits).

The 8-Bit Mode corresponds to the default state of the increment as displayed on the CRT.

i.e.: Address Increment . . . 1

8, 16 and 32-Bit Modes are the most widely used configurations, however the Zm2000 has an inbuilt flexibility whereby other bit-modes can be applied if required by the user.

8 ADDRESS INCREMENTS ARE AVAILABLE FOR SELECTION VIA THE KEYBOARD

ADDRESS INCREMENT	BIT MODE	BYTES PER WORD	No. of Devices required for Load & Program
1	**8-BIT MODE**	**1**	**1**
2	**16-BIT MODE**	**2**	**2**
3	24-BIT MODE	3	3
4	**32-BIT MODE**	**4**	**4**
5	40-BIT MODE	5	5
6	48-BIT MODE	6	6
7	56-BIT MODE	7	7
8	64-BIT MODE	8	8

3.2 8-Bit Mode (Address Increment . . . 1)

PROGRAM

Data at every location within limits that automatically default to the device size, will be programmed from the RAM to the device.

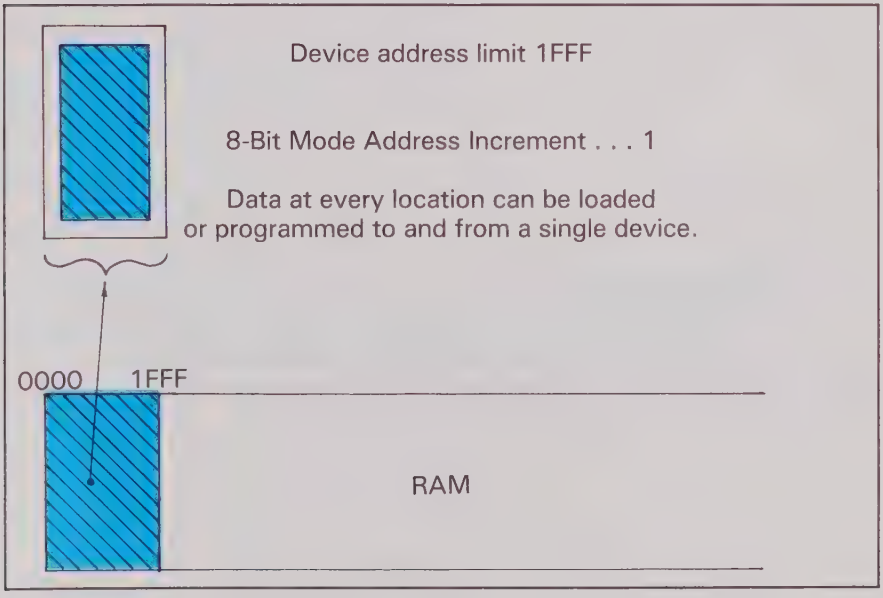
After program is complete the device will contain identical information to that of the RAM.

LOAD

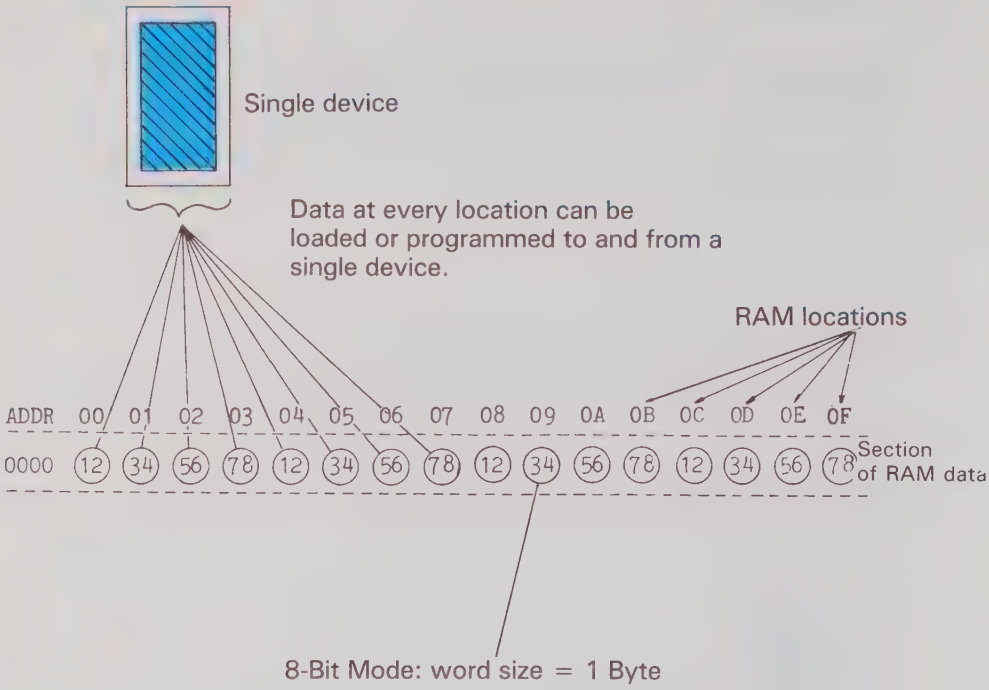
Data at every device location will be loaded into the RAM.

After load is complete the RAM will have the same address limits and will contain identical information to that of the device.

For clarity the following example will show a fixed device size:
1FFF (2764 EPROM)


Device address limit 1FFF
8-Bit Mode Address Increment . . . 1
Data at every location can be loaded
or programmed to and from a single device.
0000
1FFF
RAM

Individual bytes of data make up one word in the 8-Bit Mode:


Single device
Data at every location can be loaded or programmed to and from a single device.
RAM locations
ADDR 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
Section of RAM data
8-Bit Mode: word size = 1 Byte

VERIFY (TEST MODE)

Data transmitted can be verified with its source after either load or program is complete, providing stop and start addresses and an address increment remain consistent.

3.3 16-Bit Mode (Address Increment . . . 2)

PROGRAM

Two 8-Bit devices can be programmed in two operations from a common address range within the RAM. The address data will be 'split' during this programming sequence into two seperate bytes of data: Byte 1 and 2.

FIRST DEVICE
Address Increment . . . 2
Byte 1
RAM Start address 0000

Byte 1 will be programmed into the first device from the RAM. It represents the (MSB) ''most significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a start address of 0000 selects alternate bytes of data at even numbered locations within the RAM's address range and programs them into the device.

SECOND DEVICE
Address Increment . . . 2
Byte 2
RAM Start address 0001

Byte 2 will be programmed into the second device from the RAM. It represents the (LSB) ''least significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2 with a RAM start address of 0001 selects alternate bytes of data at odd numbered locations within the RAM's address range and programs them into the device.

LOAD

Two 8-Bit devices can load in two operations into a common address range within the RAM.

The address data to be loaded is split between the devices into two seperate bytes of data: Byte 1 and Byte 2.

FIRST DEVICE
Address Increment . . . 2
Byte 1
RAM Start address 0000

Byte 1 will be loaded from the first device, into the RAM. It represents the (MSB) ''most significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a RAM start address of 0000, selects data bytes from every 'first device' location and loads them alternately into even numbered RAM locations.

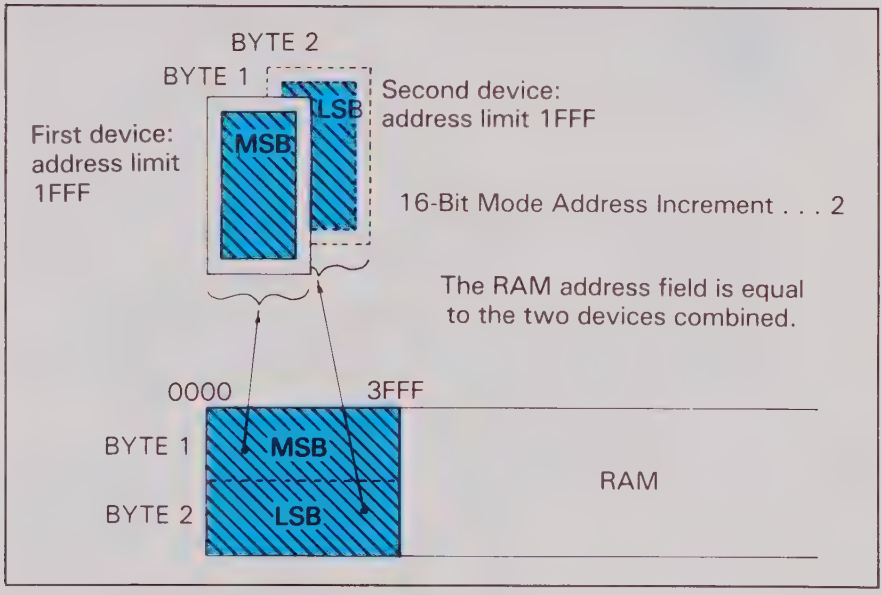
SECOND DEVICE
Address Increment . . . 2
Byte 2
RAM start address 0001

Byte 2 will be loaded from the second device, into the RAM. It represents the (LSB) ''least significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a RAM start address of 0001, selects data bytes from every 'second device' location and loads them alternately into odd numbered RAM locations.

A graphic example of how the 16-Bit Mode (Increment . . . 2) works on both LOAD and PROGRAM is shown below:

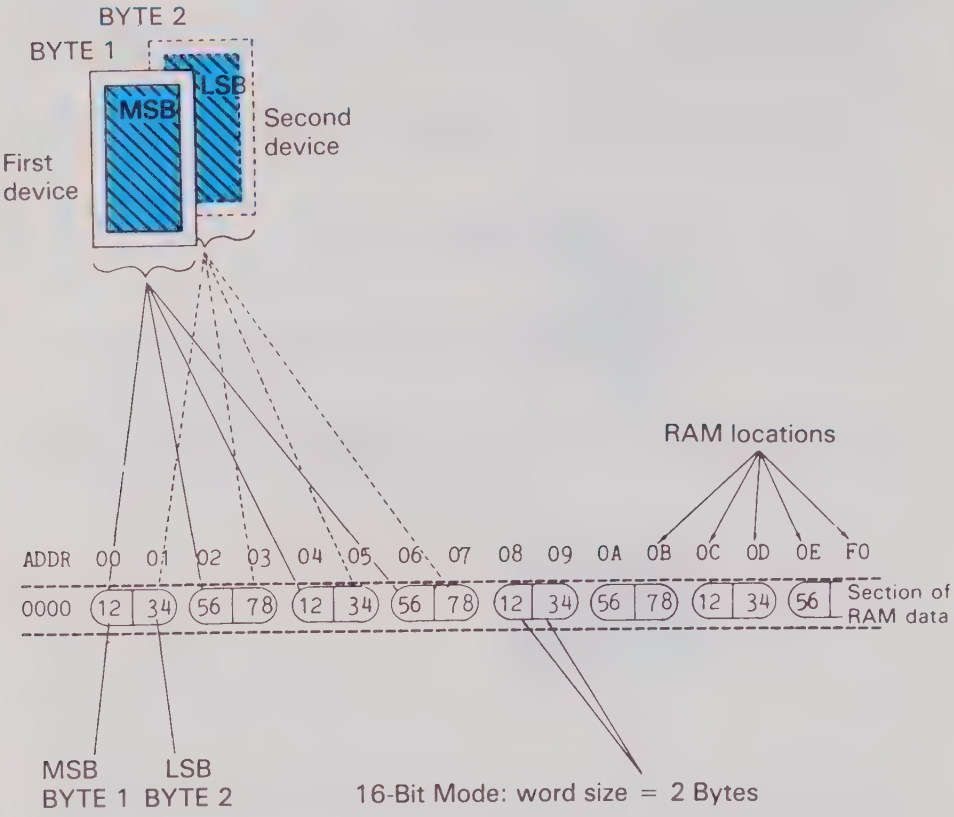
For clarity the following example will show a fixed device size: 1FFF (2764 EPROM)



In the first operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only the most significant byte "MSB" of a 16-Bit Word undergoing data transfer.

In the second operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only the least significant byte "LSB" of a 16-Bit word undergoing data transfer.

Two bytes of data make up one word in the 16-Bit Mode:


BYTE 2
BYTE 1
LSB
MSB
Second device
First device
RAM locations
ADDR 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E F0
0000 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56
Section of RAM data
MSB BYTE 1
LSB BYTE 2
16-Bit Mode: word size = 2 Bytes

MSB—The "most significant BYTE" in binary code
LSB—The "least significant BYTE" in binary code

VERIFY (TEST MODE)

Data transmitted can be verified with its source after both load or program are complete, providing that stop and start addresses and an address increment remain consistent.

3.4 32-Bit Mode (Address Increment . . . 4)

PROGRAM

Four 8-Bit devices can be programmed in four operations from a common address range within the RAM. The address data will be divided during this programming sequence into four seperate bytes of data: Byte 1, Byte 2, Byte 3 and Byte 4.

FIRST DEVICE
Address Increment . . . 4
Byte 1
RAM Start Address 0000

Byte 1 will be programmed into the first device from the RAM. It represents the (MSB) ''most significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a RAM start address of 0000 selects every fourth byte of data within the RAM's address range (0000, 0004, 0008, 000C etc) and programs them into the device.

SECOND DEVICE
Address Increment . . . 4
Byte 2
RAM Start Address 0001

Byte 2 will be programmed into the second device from the RAM. It represents the 'second most significant byte' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0001 selects every fourth byte of data within the RAM's address range (0001, 0005, 0009, 000D etc) and programs them into the device.

THIRD DEVICE
Address Increment . . . 4
Byte 3
RAM Start Address 0002

Byte 3 will be programmed into the third device from the RAM. It represents the ''second least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0002 selects every fourth byte of data within the RAM's address range (0002, 0006, 000A, 000E etc) and programs them into the device.

FOURTH DEVICE
Address Increment . . . 4
Byte 4
RAM Start Address 0003

Byte 4 will be programmed into the fourth device from the RAM. It represents the (LSB) ''least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0003 selects every fourth byte of data within the RAM's address range (0003, 0007, 000B, 000F etc) and programs them into the device.

Four 8-Bit devices can load in four operations from a common address range within the RAM.

The address data to be loaded is divided between the devices into four seperate bytes of data: Byte 1, Byte 2, Byte 3 and Byte 4.

FIRST DEVICE
Address Increment . . . 4
Byte 1
RAM Start Address 0000

Byte 1 will be loaded from the first device, into the RAM. It represents the (MSB) ''most significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a RAM start address of 0000 selects data bytes from every 'first device' location and loads them into the RAM at every fourth location: 0000, 0004, 0008, 000C etc.

SECOND DEVICE
Address Increment . . . 4
Byte 2
RAM Start Address 0001

Byte 2 will be loaded from the second device, into the RAM. It represents the 'second most significant byte' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0001 selects data bytes from every 'second device' location and loads them into the RAM at every fourth location: 0001, 0005, 0009, 000D, etc.

THIRD DEVICE
Address Increment . . . 4
Byte 3
RAM Start Address 0002

Byte 3 will be loaded from the third device, into the RAM. It represents the ''second least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0002 selects data bytes from every 'third device' location and loads them into the RAM at every fourth location: 0002, 0006, 000A, 000E etc.

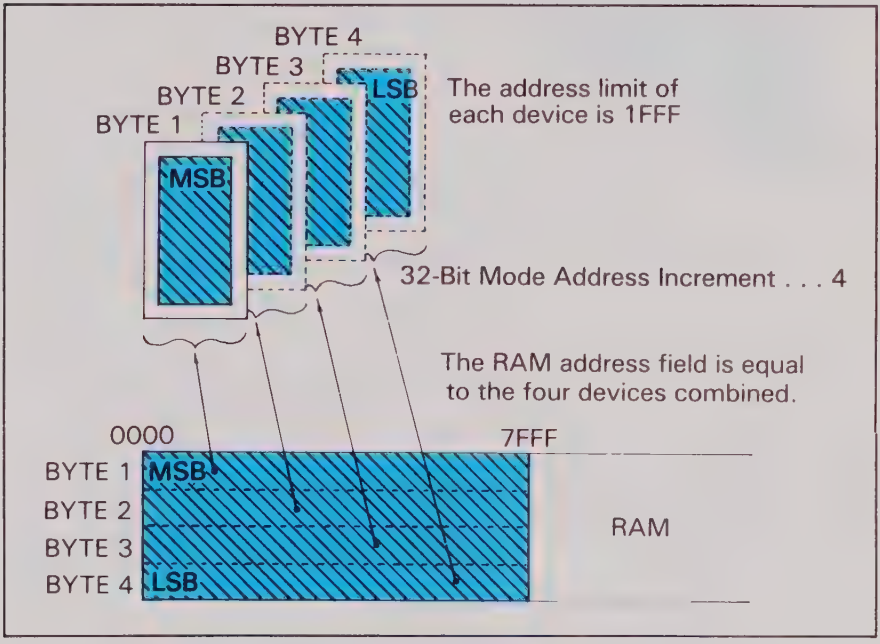
FOURTH DEVICE
Address Increment . . . 4
Byte 4
RAM Start Address 0003

Byte 4 will be loaded from the fourth device into the RAM. It represents the (LSB) ''least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0003 selects data bytes from every 'fourth device' and loads them into the RAM at every fourth location: 0003, 0007, 000B, 000F etc).

A graphic example of how the 32 Bit Mode (increment . . . 4) works on both LOAD and PROGRAM is shown below:

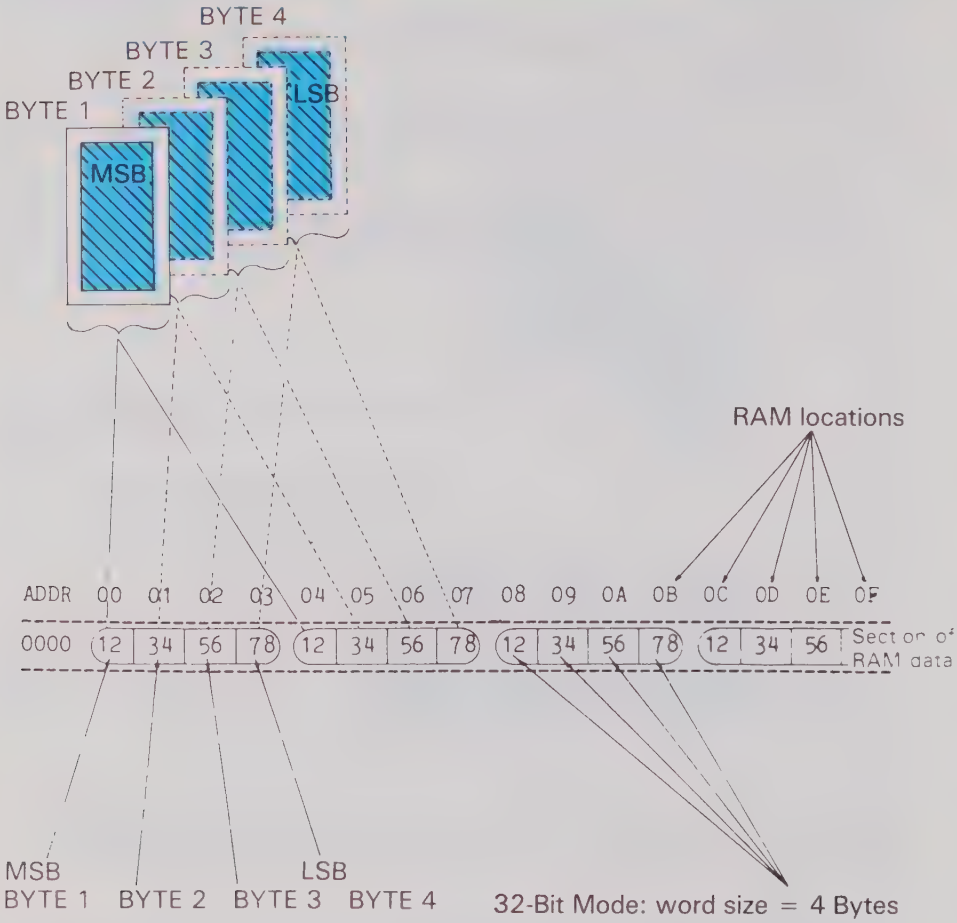
For clarity the following example will show a fixed device size: 1FFF (2764 EPROM)



In the first operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only BYTE 1 the "most significant byte" of a 32-Bit Word undergoing data transfer.

This sequence is repeated for the next three devices with bytes 2, 3 and 4 respectively undergoing data transfer.

Four bytes of data make up one word in the 32-Bit Mode:



MSB—The "most significant BYTE" in binary code
LSB—The "least significant BYTE" in binary code

VERIFY (TEST MODE)

Data transmitted can be verified with its source after both LOAD or PROGRAM are complete, providing stop and start addresses and an address increment of 4 remain consistent with those specified in either operation.

EMPTY TEST

The "address increment" feature, is included in the EMPTY TEST Mode. However, as an EMPTY TEST is applied to the device alone, the "increment function" plus "RAM start address", become superfluous features.

3.5 NIBBLE SWAP (Increment on 4-Bit devices)

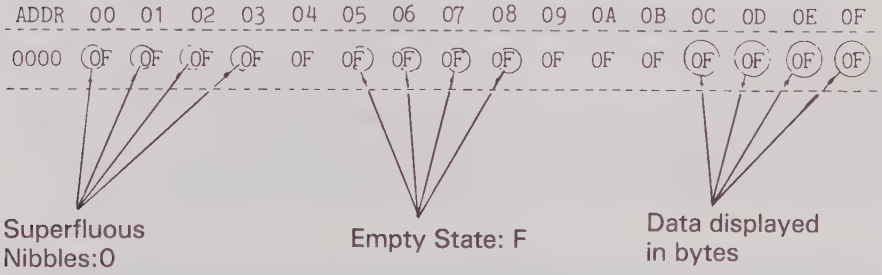
Two 4-Bit devices can be programmed in two operations from a common address range within the RAM, in such a way as to contain 8 Bit Data when combined.

The address data will be 'split' during this programming sequence into two seperate nibbles of data.

4-BIT DEVICE. INITIAL DISPLAY

The 4-Bit data within a particular device can be displayed (with the device inserted into its appropriate ZIF socket) by pressing Key 6.

This example shows a section of the display from locations 0000 to 000F.


ADDR 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
Superfluous Nibbles:0
Empty State: F
Data displayed in bytes

As can be seen from the above section, the data displayed in each individual location is in the form of bytes. However, the left nibbles containing 'O's' are superfluous throughout the entire device address range. Only the right nibble reflects the condition of the 4-Bit device. In this case the empty state—'F'.

N.B.—In other examples the empty state may be '0' in which case both nibbles would look alike however, the above conditions would still apply.

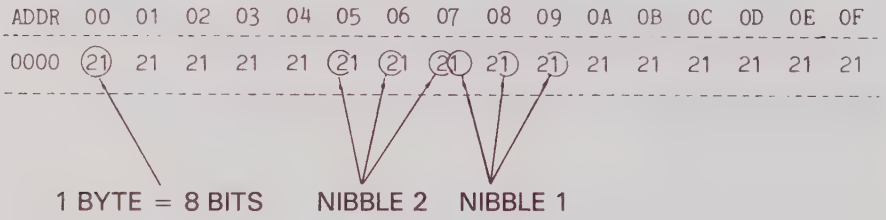
PROGRAMMING OF 8 BIT DATA INTO TWO 4 BIT DEVICES

8-Bit data can be loaded into the RAM in one of three ways:

i From peripheral equipment via the interface ports.
ii From a master 8-Bit Device.
iii By use of the Edit functions.

By pressing key 5 the 8-Bit RAM data can be displayed.

This example shows a section of RAM, from locations 0000 to 000F.



Two 4-Bit devices can be programmed in two operations from a common address range within the RAM. The address data will be 'split' during this programming sequence into two seperate nibbles of data: Nibble 1 and Nibble 2.

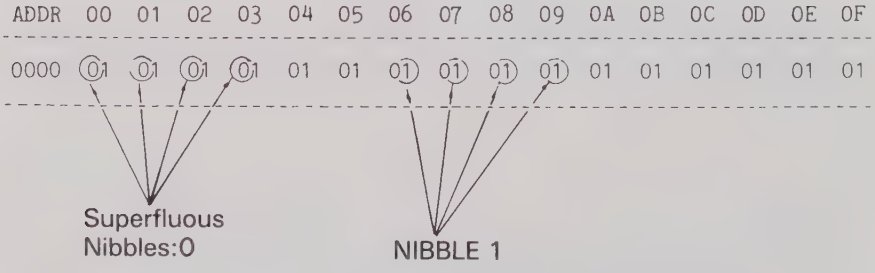
For clarity the following examples will show a fixed 4-Bit device size: 00FF (Bipolar PROM) and will be without preselected program offsets.

FIRST 4-BIT DEVICE
NIBBLE 1

Only the right hand nibble of each data byte will be selected. The RAM will automatically program these nibbles into every location of the device.

When the program is complete the contents of the first device can be displayed by pressing RESET followed by KEY . . . 6

Section of display 0000 to 000F (first device)



Before the second device can be programmed, the RAM data within the specified limits of 0000 to 00FF must first undergo NIBBLE SWAP:

To enter NIBBLE SWAP:

PRESS KEY 3

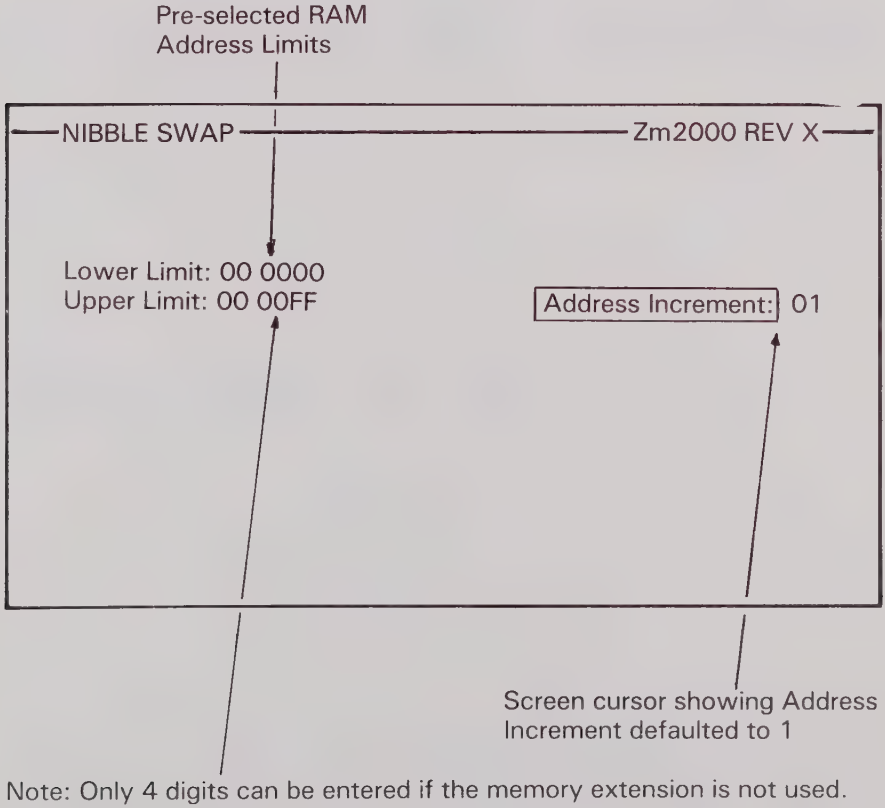
This will display the EDIT RAM DATA sub-menu, then

PRESS KEY 6

The screen will now display the default states of the RAM address limits to undergo the nibble swap. The screen cursor will show the lower RAM address limit defaulted to zero, and the upper limit will be defaulted to the size of the pre-selected device. Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓), and the new selection can be entered from the hex-keyboard. To confirm these selections:

PRESS ENTER

This will display the following:



Note: Only 4 digits can be entered if the memory extension is not used.

If required by the user, the nibble swap function on 4-bit devices can be integrated with Address Increment to handle data in the 16-bit mode and beyond.

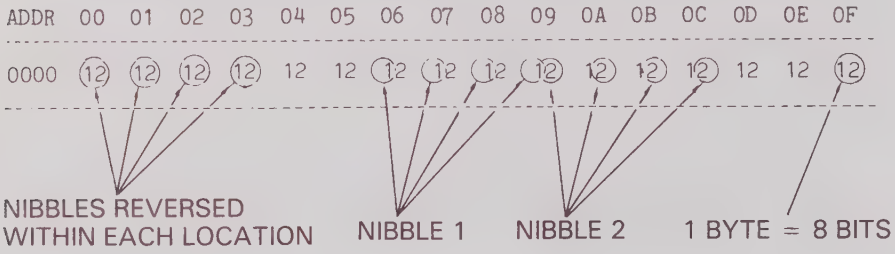
The Address Increment can take values from 1 to 8 which can be entered from the keyboard. To confirm the Address Increment value and implement the nibble swap:

PRESS ENTER

To display rearranged RAM data:

PRESS RESET followed by KEY 5

Section of RAM data 0000 to 000F



To return to the SYSTEM MENU:

PRESS RESET

The second 4-bit device can now be programmed.

SECOND 4-BIT DEVICE

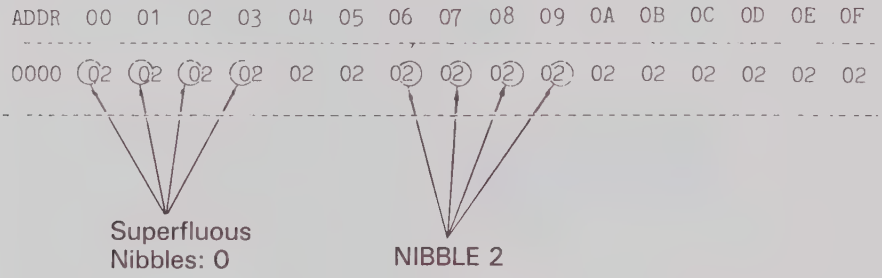
NIBBLE 2

Replace the first device in the ZIF socket with the second 4-bit device to be programmed.

Once again only the right hand nibble of each data byte will be selected. The RAM will automatically program these nibbles into every location within the device.

When the program is complete the contents of the second device can be displayed by pressing RESET followed by KEY . . . 6.

Section of display 0000 to 000F second device



To return to the SYSTEM MENU:

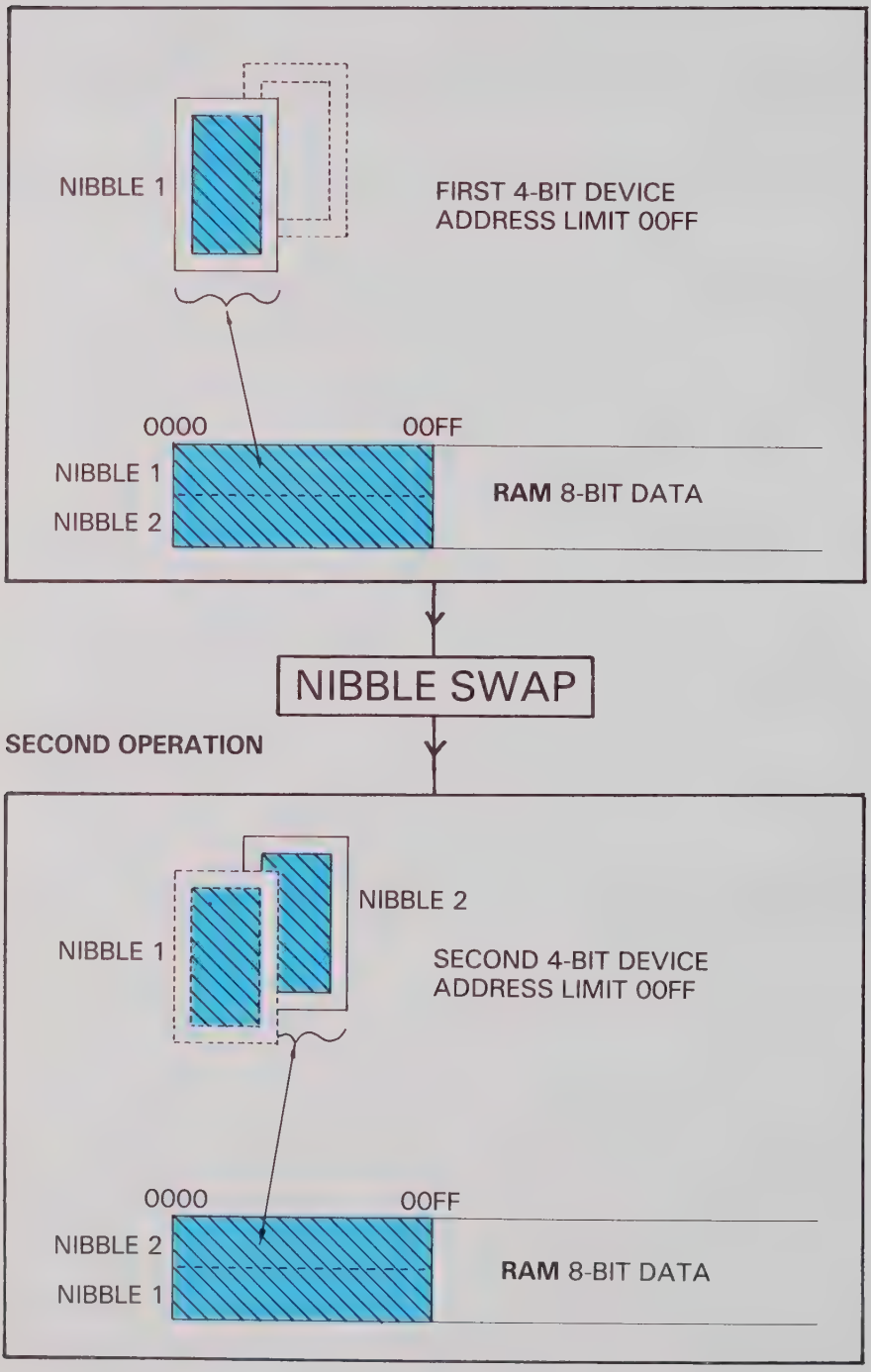
PRESS RESET

LOAD

Loading of two 4-Bit devices into the RAM is a direct reversal of the programming sequence.

INCREMENT ON TWO 4-BIT DEVICES BY USE OF THE NIBBLE SWAP FUNCTION

FIRST OPERATION



SECTION 4

stag

Sophisticated systems for the discerning engineer.

4.—DEVICE FUNCTIONS

4.1 LOAD

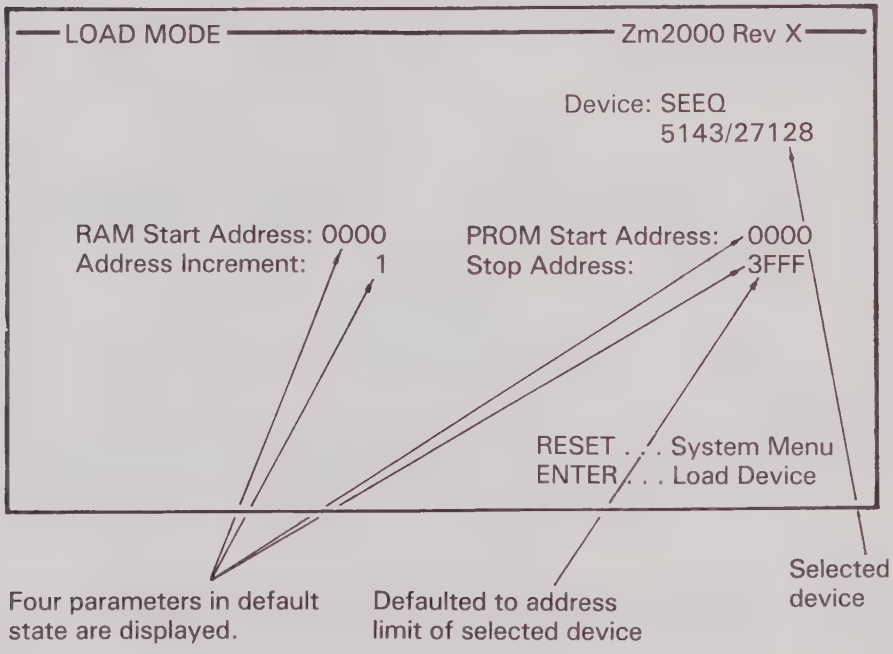
Loading the RAM from a master device.

Pressing RESET will return any pre-selected mode to the SYSTEM MENU.

Insert the master device into the appropriate ZIF socket as indicated by the red LED adjacent to Pin 1.

To implement the LOAD:

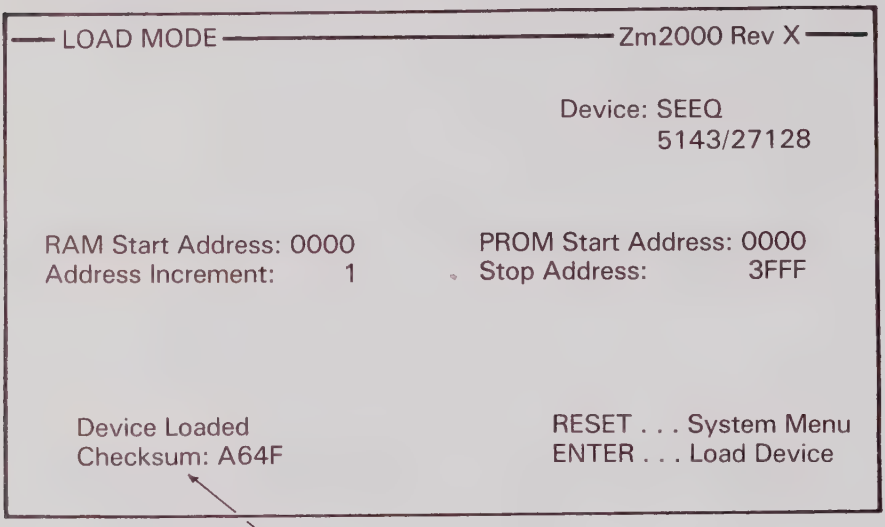
PRESS LOAD



These parameters define the address limits for loading the device data into the RAM.

If stop and start addresses are not required then pressing the 'ENTER' key will implement LOAD

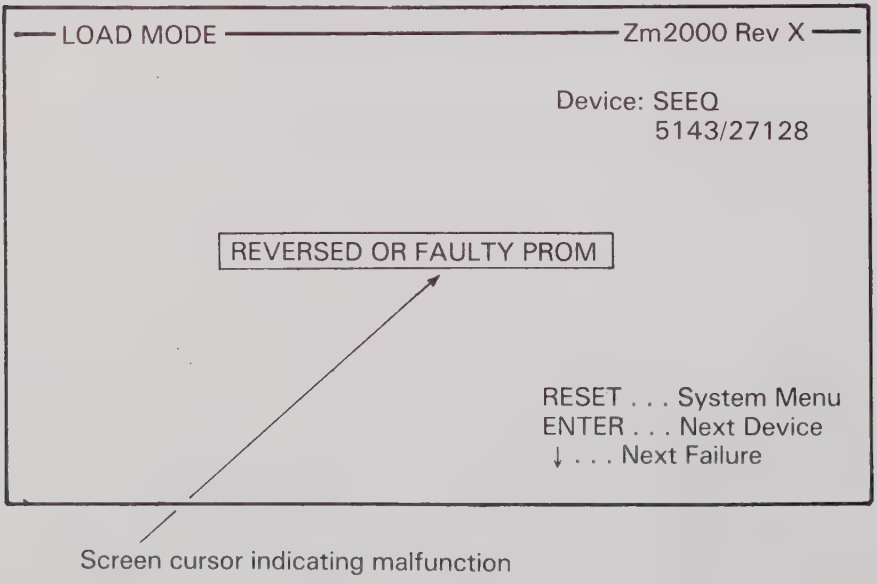
During load, a two digit address counter will increment to show that LOAD is in progress.



On completion the display will show a checksum of all data LOADED into the RAM.

LOAD pre-checks are carried out automatically and any malfunctions will be displayed for faulty or incorrectly socketed parts.

For example:



LOAD: Device/RAM address limits

Any of the displayed settings may be altered from their default state if required by the user.

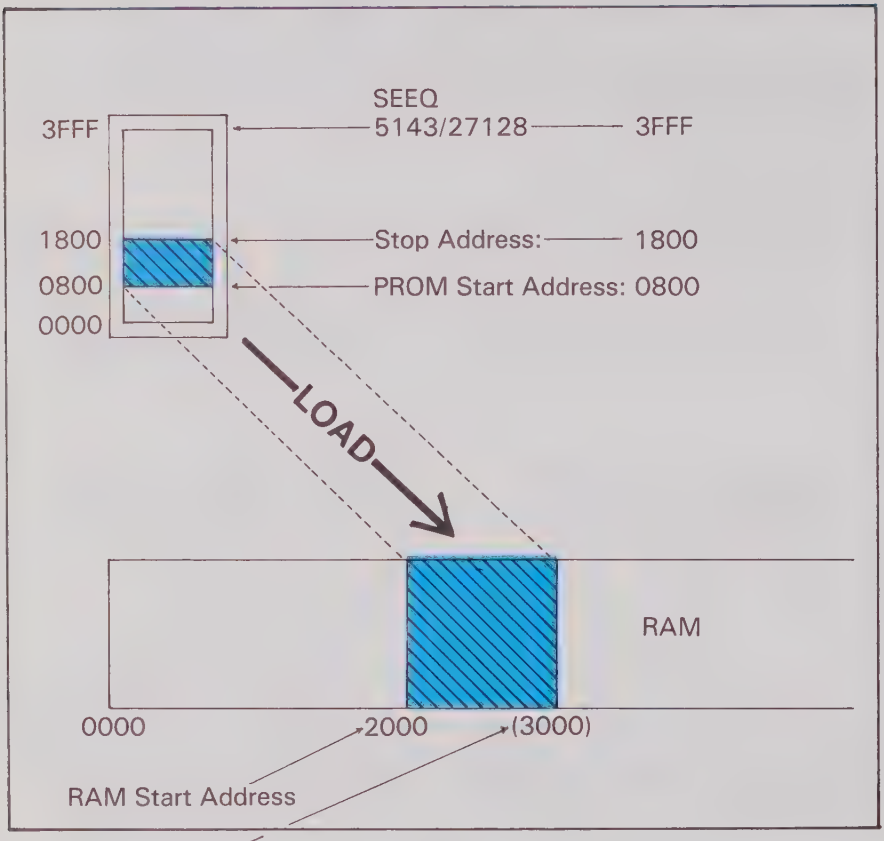
This is done by (i) use of the cursor keys to select the parameter as indicated by a flashing cursor and (ii) by use of the hex-keyboard to enter the new address.

For example:

```
—LOAD MODE———————————————————— Zm2000 Rev X—

                                  Device: SEEQ
                                          5143/27128

RAM Start Address: 2000          PROM Start Address: 0800
Address Increment:    1          Stop Address      : 1800

                                  RESET . . . System Menu
                                  ENTER . . . Load Device
```

For clarity the Address Increment . . . 1 will remain in the default state (8-Bit Mode).

Examples of the address limits for the master device and the RAM can be shown in diagram form:



No stop address is entered for the RAM. Data loaded will automatically default to the size of the data block specified within the device.

4.2 TEST OPERATIONS (Program Sequence)

The program sequence incorporates all the various test modes that may be required by the user prior to, during and after programming of a device.

The device tests fall into three categories:

(i) MANUAL

EMPTY and VERIFY—using the TEST key.

(ii) OPTIONAL PRE-PROGRAM

ILLEGAL BIT, EMPTY or NONE—select in SET SYSTEM mode.

(iii) AUTOMATIC IN-PROGRAM

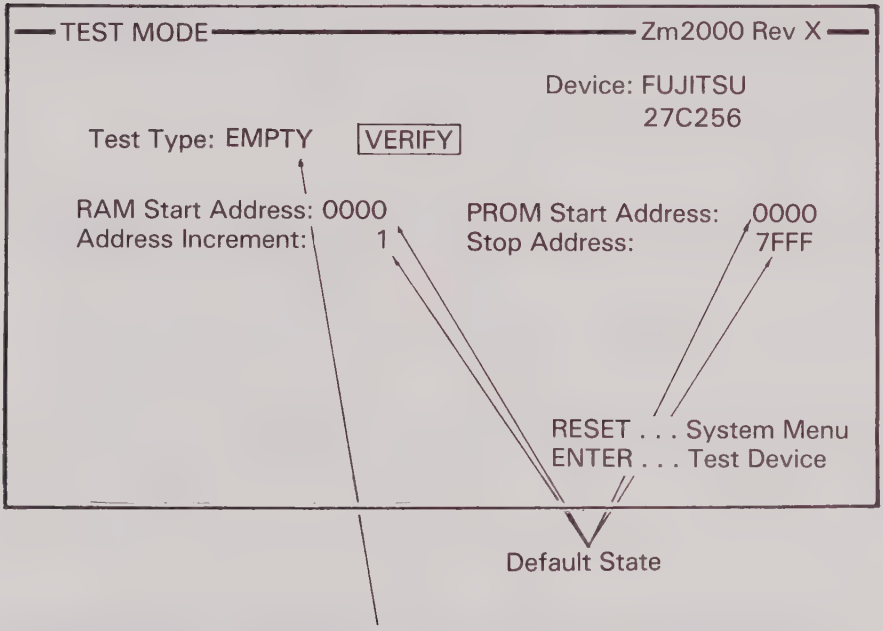
VERIFY, PROGRAM FAIL and CHECKSUM—always performed at the end of a device PROGRAM operation.

4.3 EMPTY TEST (Manual)

An EMPTY TEST can be applied to the device in the appropriate ZIF socket at any time. To do this:

PRESS TEST

This will show the TEST MODE display:



Either the VERIFY or the EMPTY TEST will be indicated by a flashing cursor. Interchange between these two test modes is made by use of the horizontal cursor keys.

If the EMPTY TEST is applied to the whole device then stop and start addresses can remain in their default state as above.

To implement the EMPTY TEST:

PRESS ENTER

The device will be examined for its unprogrammed state throughout the specified area as defined by the start and stop addresses. The unprogrammed state could be either 'FF' or '00' for an 8-bit device or 'F' or '0' for a 4-bit device.

During the EMPTY TEST a two digit counter will increment to show that the operation is in progress.

On completion the display will show a checksum

For example:

```
— TEST MODE ——————————————————— Zm2000 Rev X —

                                    Device: FUJITSU
                                            27C256
Test Type: [EMPTY]  VERIFY

   RAM Start Address: 0000    PROM Start Address: 0000
   Address Increment:    1    Stop Address:       7FFF




Device Empty                   RESET . . . System Menu
Checksum: 8000                 ENTER . . . Test Device
```

To return to the SYSTEM MENU:

PRESS RESET

EMPTY TEST: FAIL

If a device were to fail the EMPTY TEST the display would show:

i. up to eight locations where the discrepancy occurs (these will be the first eight locations in the device where data was detected).
ii. the unblown state of that device.
iii. the actual data within each of the device locations.

For example:

TEST MODE — Zm2000 Rev X

Device: FUJITSU
27C256

Address	Empty State	PROM Data
0047	FF	AC
0084	FF	BD
0164	FF	F3
034A	FF	12
035A	FF	02
0597	FF	88
05DD	FF	CE
0600	FF	44

RESET . . . System Menu
ENTER . . . Test Device
↓ . . . Next Failure

Not Empty

Pressing the down cursor key will show the next fail locations if any are present. By continually pressing this key the user can scan every location in the device.

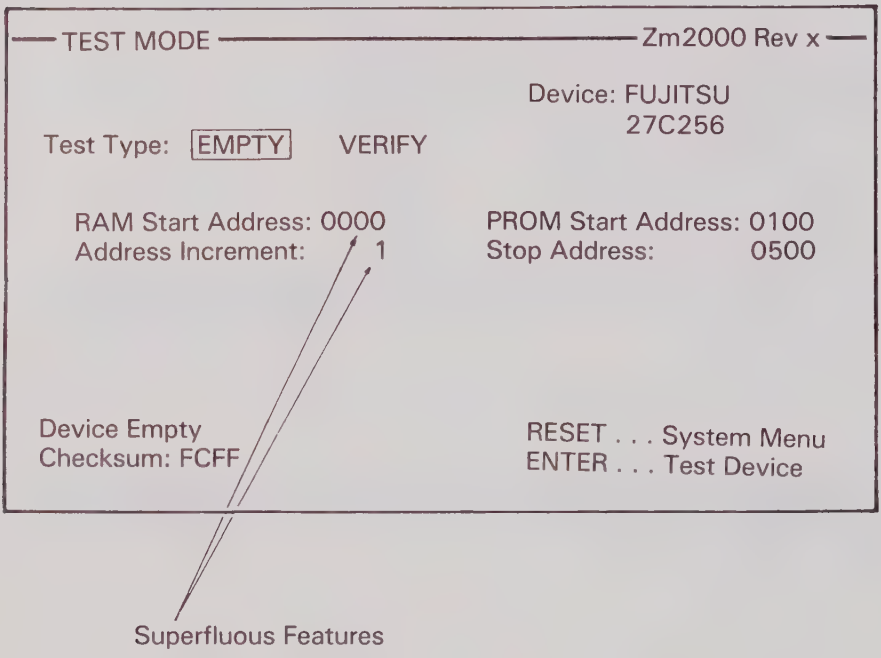
EMPTY TEST: Stop and Start Address

The EMPTY TEST is applied to the device alone therefore, the ''Address Increment'' and the ''RAM Start Address'' are superfluous features.

Only the ''PROM Start Address'' and ''Stop Address'' are relevant. In this way the 'empty state' scan of the PROM can be made within pre-selected limits.

These can be entered by use of the cursor keys to select the parameters and by use of the hex-keyboard to enter the new address.

For instance:



To implement the EMPTY TEST:

PRESS ENTER

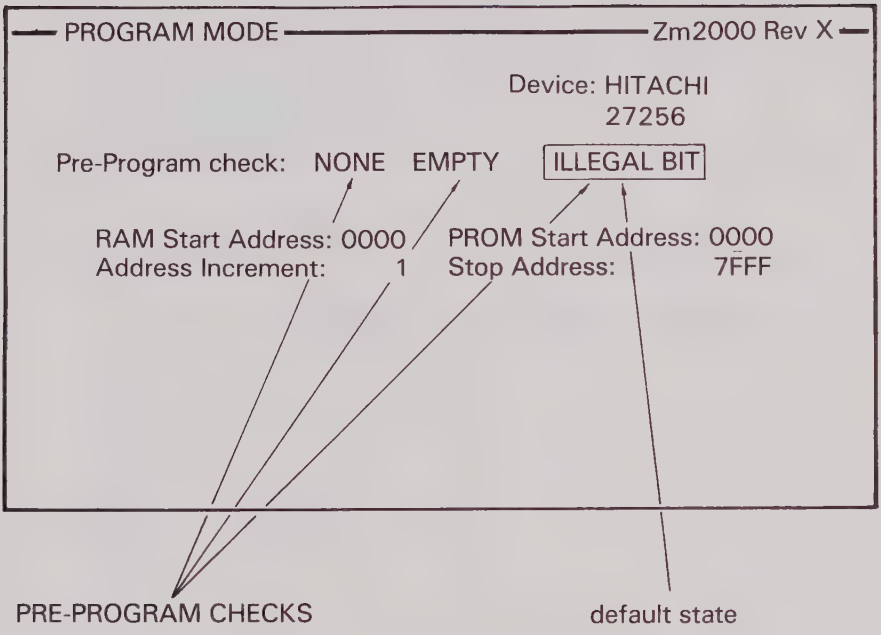
To return to the System Menu:

PRESS RESET

4.4 PRE-PROGRAM CHECKS

To enter the PROGRAM mode:

PRESS 'PROG' to display



PRE-PROGRAM CHECKS are an automatic scan of the device prior to programming. The checks available are:

i ILLEGAL BIT TEST (default state)
ii EMPTY TEST
iii a third option where no check at all is applied, i.e. NONE

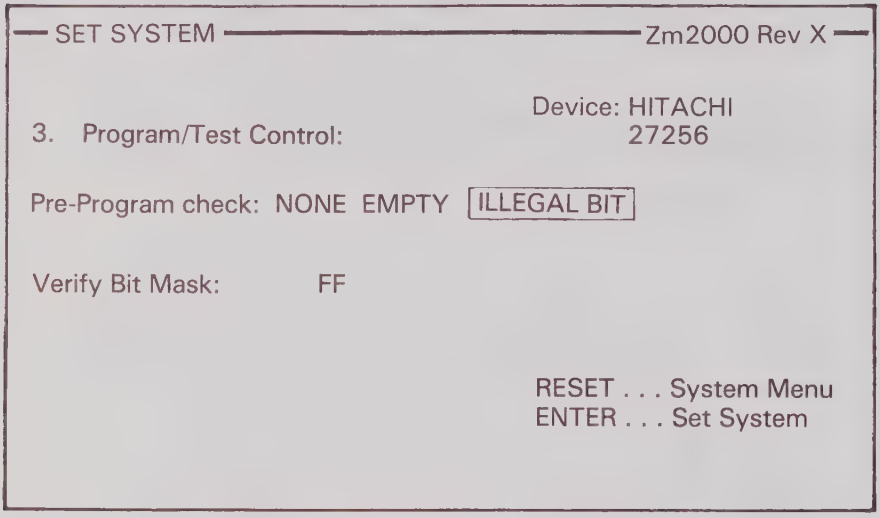
There is also a check for faulty or incorrectly socketed parts. Should a fail of this nature occur, the message ''Reversed or Faulty Prom'' will appear on the screen and the programming will be aborted.

The ILLEGAL BIT CHECK is the default state (as indicated by the screen cursor) therefore the device will automatically undergo this pre-program check unless either of the other two options is selected.

To make this selection:

PRESS RESET (to return to the SYSTEM MENU)

PRESS KEY A followed by KEY 3



The selection can now be made using the horizontal cursor keys. To confirm the selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

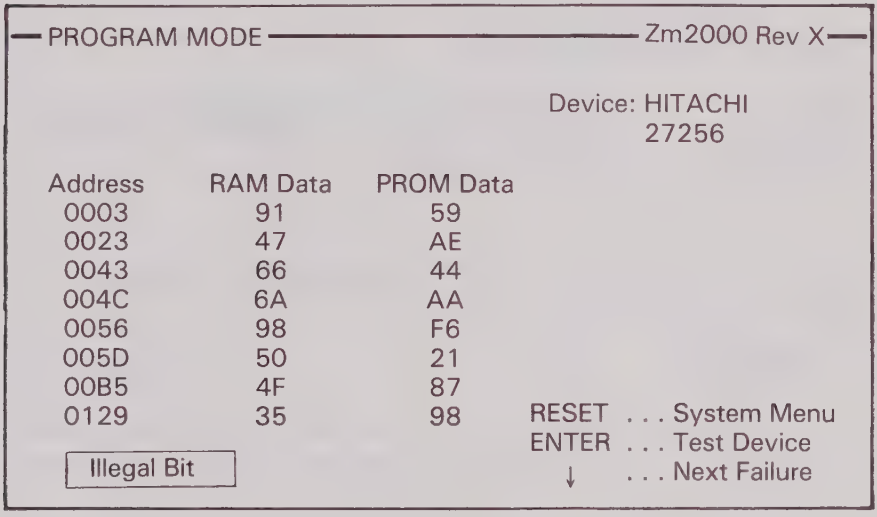
ILLEGAL BIT: PRE-PROGRAM CHECK

The PPZ automatically checks that the pattern already within the device is able to be programmed with the intended data from the RAM.

If a device were to fail the ILLEGAL BIT TEST the display would show:

i. up to eight locations where this discrepancy occurs (these will be the first eight locations in the device where a fault is detected).
ii. the data from the RAM destined for each location and
iii. the data within the device that has caused the failure.

For example:



Pressing the down cursor key will show the next fail locations if any are present. By continually pressing this key the user can scan every location in the device.

EMPTY: PRE PROGRAM CHECK

This is an automatic scan to ensure the EMPTY STATE of the device prior to programming.

It is basically the same as the previously described 'manual' empty test but differs in two ways:

i. the device will automatically undergo programming should it pass the test, therefore no checksum of the unblown state is displayed and

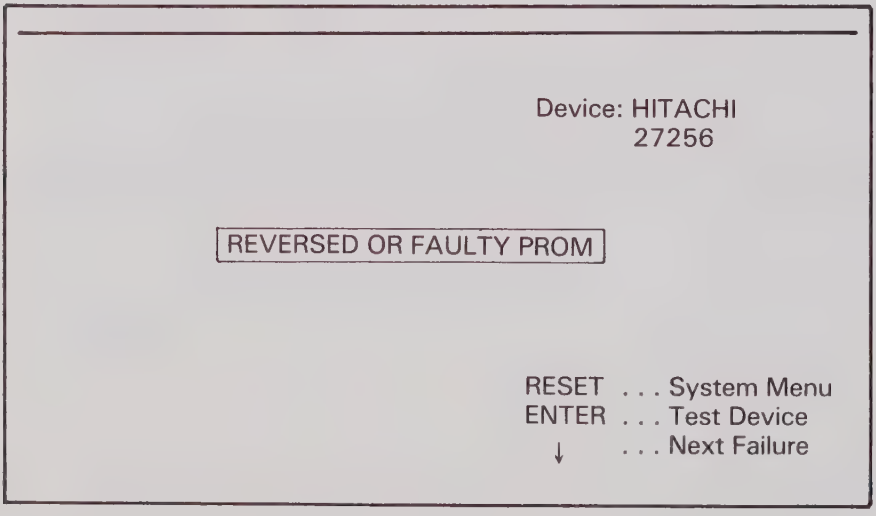
ii. the stop and start addresses are dictated by those entered in the PROGRAM MODE.

NONE: NO PRE-PROGRAM CHECK

The third option is a lack of any automatic pre-program checks. However a 'PROGRAM FAIL' will still be indicated if incompatible data is received by the device at a programmable location.

If a part to be programmed is faulty or badly socketed, the display will indicate the malfunction.

For example:

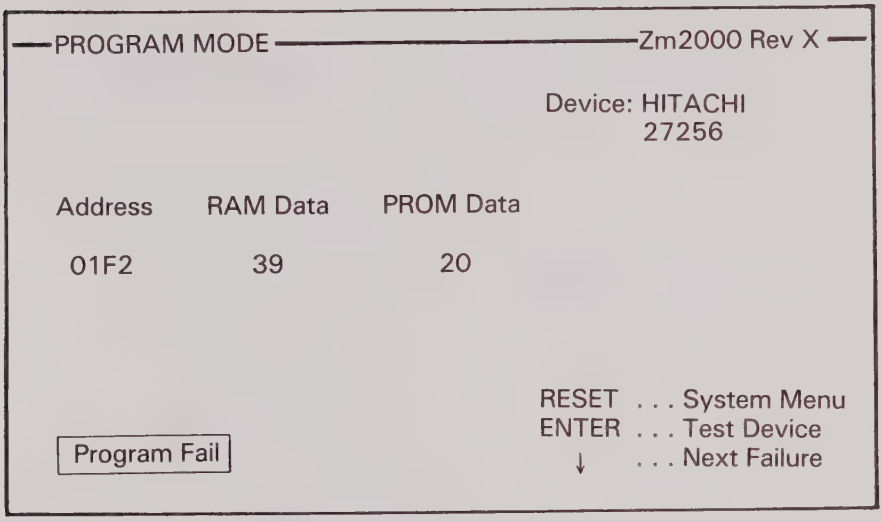


PROGRAM FAIL

If a PROGRAM FAILURE occurs the display will show:

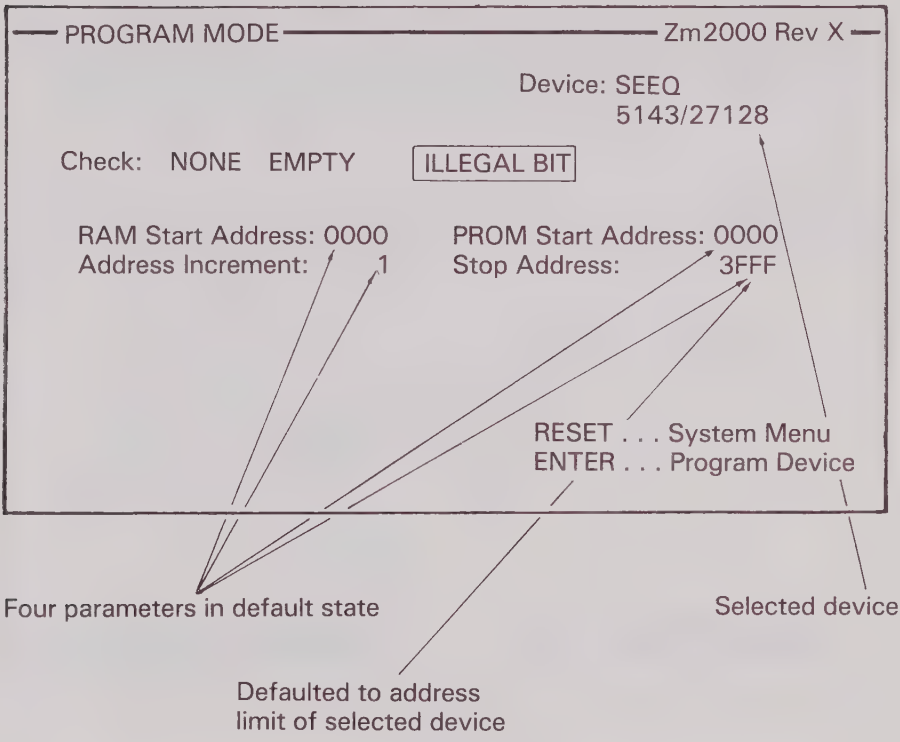
i. the first location where a discrepancy has occured (unlike other fails only one location at a time will be displayed).
ii. the data within the RAM at that location and
iii. the data within the PROM at that location

For example:



4.5 PROGRAMMING

After selection of the PRE-PROGRAM CHECK in the SET SYSTEM mode, the PROGRAM mode is entered by pressing the PROG key.

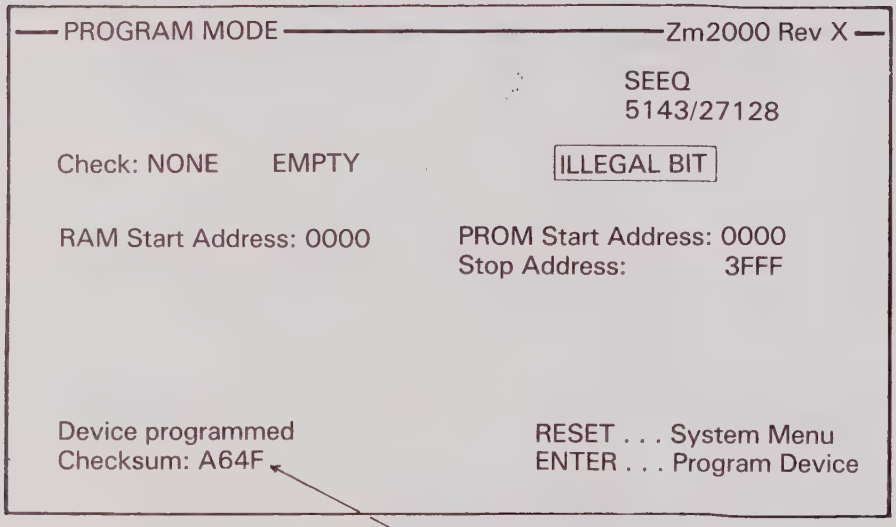


These parameters define the address limits for programming the RAM data into the device.

If stop and start addresses are not required then to implement the PROGRAM operation:

PRESS ENTER

During programming a two digit address counter will increment to show that firstly a pre-program check is in progress, secondly the program itself is in progress and thirdly an automatic in-program verify is in progress. Each of these counts will start and stop within the selected address range.



On completion the display will show a checksum of data programmed into the device.

If the data to be programmed into a particular location is the same as the unblown state of that device, the programming sequence will automatically skip to the next location where information is stored. This function speeds up programming considerably where large sections of the device are to remain empty.

To abort programming:

PRESS RESET

Note: Some programming algorithms require the key to be held down for several seconds before the operation is terminated.

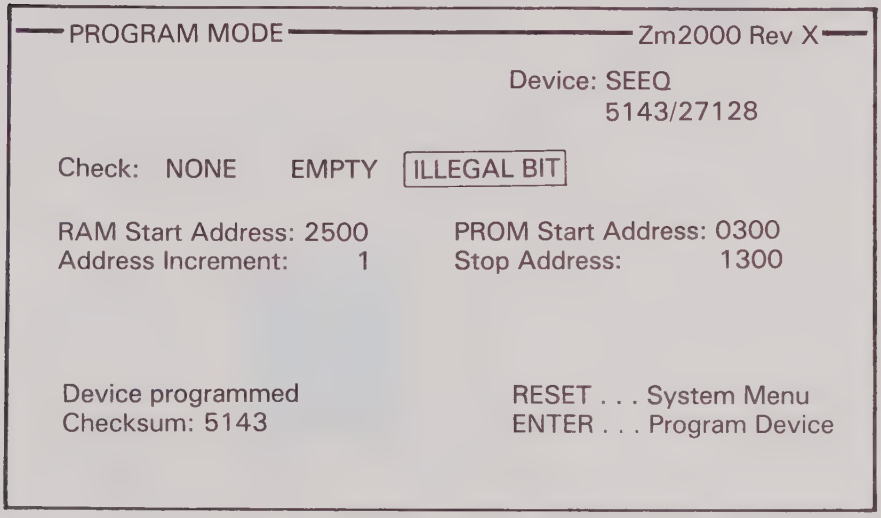
PROGRAM: Device/RAM address limits

Any of the displayed settings may be altered prior to programming from their default state if required by the user.

This is done by:

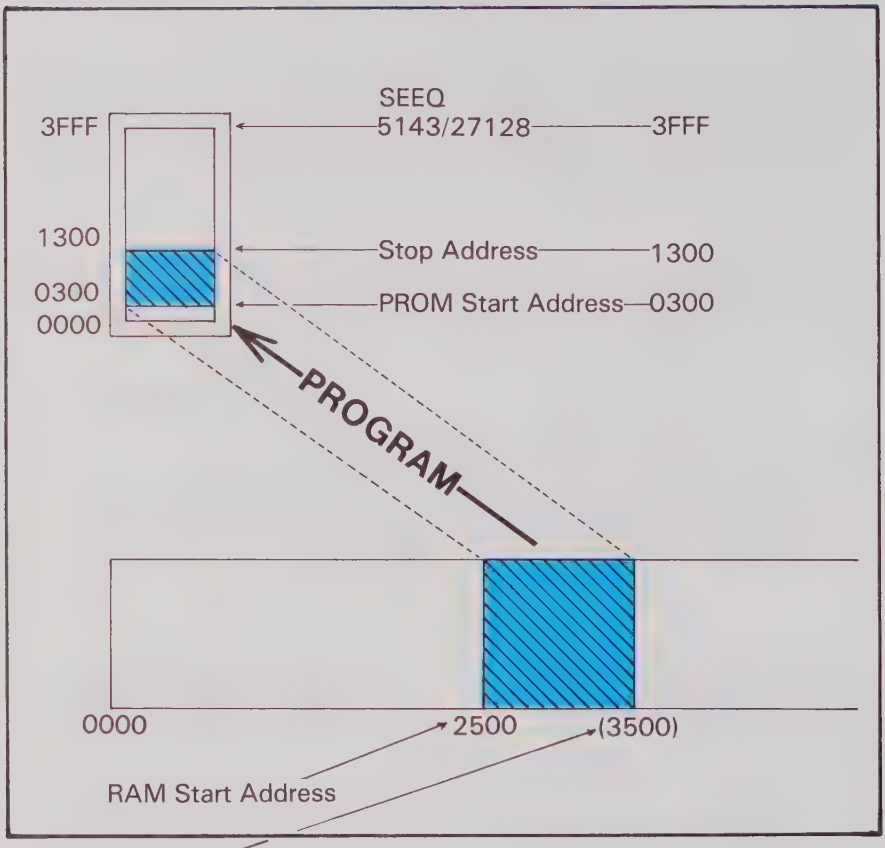
i. use of the cursor keys to select the parameter as indicated by a flashing cursor
ii. by use of the hex-keyboard to enter the new address.

For example:



For clarity the Address Increment . . . 1 will remain in the default state (8-Bit Mode)

Example of the address limits for the master device and the RAM can be shown in diagram form:



No stop address is entered for the RAM as data programmed will automatically default to the size of the data block specified within the device.

IN-PROGRAM VERIFY

A feature whereby each device location programmed is immediately checked to see that it is identical to the incoming data bytes in the RAM. However, only after the two digit program counter has incremented over the entire pre-selected address range will VERIFY FAIL data be displayed.

If this were to happen the display would show:

i up to eight locations in the device where a discrepancy has occured. These will be the first eight locations where the device and the RAM data are incompatible.
ii the data from the RAM in comparison to data in the device locations and
iii the data in the device locations.

For instance:

PROGRAM MODE — Zm2000 Rev X

Device: FUJITSU
27128

Address	RAM Data	PROM Data
0500	FF	00
0501	FF	00
0502	FF	00
0503	FF	00
0504	FF	00
0505	FF	00
0506	FF	00
0507	FF	00

Verify Fail

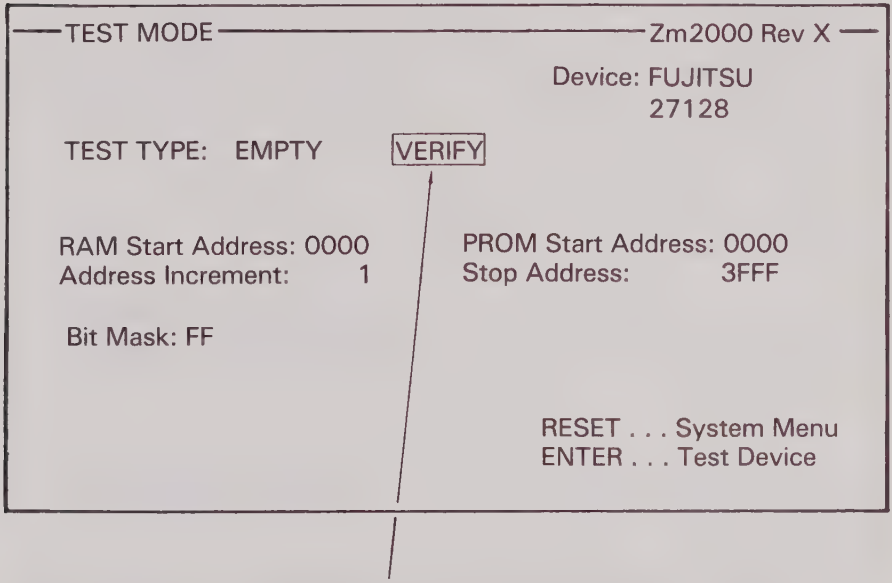
RESET . . . System Menu
ENTER . . . Test Device
↓ . . . Next Failure

Pressing the down cursor key will re-start PROGRAMMING from a new start address defaulted to the next location above the last found VERIFY FAIL. Only after the counter has fully incremented over the remainder of the pre-selected address field will any new VERIFY FAILS be displayed.

4.6 VERIFY TEST (Manual)

The manual VERIFY TEST can be applied at any time, e.g. after LOAD as well as PROGRAMMING.

It can be entered by first pressing the TEST key to show the TEST MODE display:



Either the EMPTY or the VERIFY test will be indicated by a flashing cursor. Interchange between these two modes is made by use of the horizontal cursor keys.

The manual verify is basically the same as the previously described in-program verify but differs in three ways:

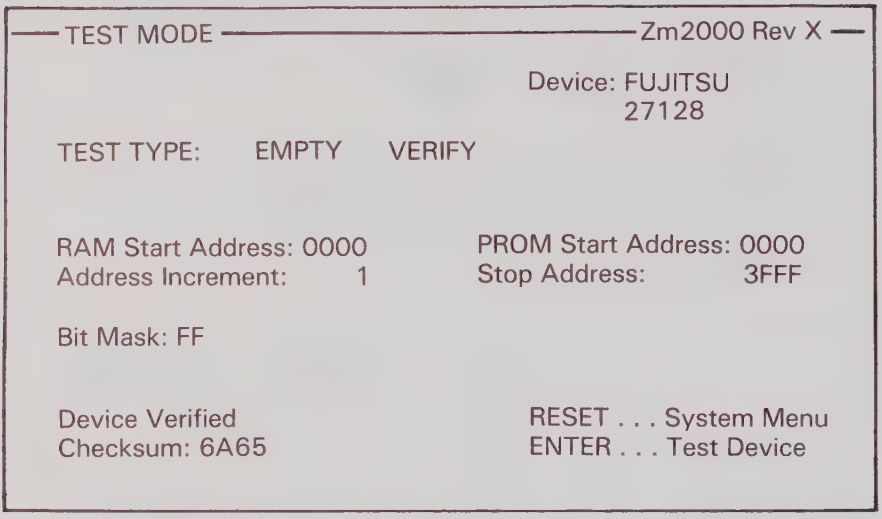
i independent stop and start addresses and an address increment can be entered for the test, whereas the in-program version defaults to limits set in the program mode.
ii manual verify is removed from the program sequence therefore scanning of the displayed RAM/device discrepancies is a much faster operation and
iii a BIT MASK can be entered.

To implement the VERIFY TEST:

PRESS ENTER

During the check a two digit counter will increment to show that the operation is in progress.
On completion the display will show a checksum of all verified data within the device.

For example:



VERIFY TEST: Address increment and stop and start addresses.

These can be entered prior to the test by use of the cursor keys to select the parameters and by use of the hex-keyboard to enter the new address.

VERIFY TEST: FAIL

If a verify fail were to occur the display would show similar information to that of the in-program verify:

TEST MODE — Zm2000 Rev X

Device: FUJITSU
27128

Address	RAM data	PROM Data
0030	A6	FF
0033	F6	FF
0053	26	22
0056	19	11
00B5	36	BC
00BA	68	E8
00BF	68	D4
0104	6E	66

Verify Fail

RESET . . . System Menu
ENTER . . . Test Device
↓ . . . Next Failure

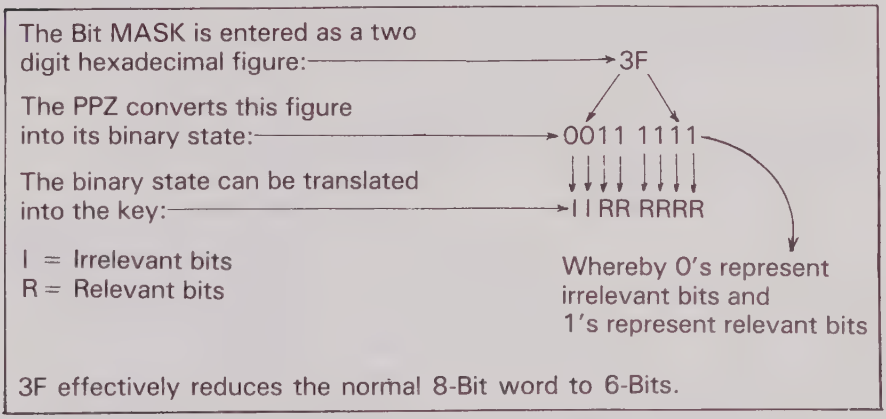
Continually pressing the down cursor key will implement this scan of the entire device address range.

VERIFY "BIT MASK"

This is a feature whereby one or more bits of the 8 bit data can be masked (made irrelevant) during a device/RAM VERIFY scan of what would normally be 8-Bit words.

The BIT MASK is entered as a two digit hexadecimal number, which is converted by the PPZ into its fundamental binary state. This can be translated as the BIT MASK key.

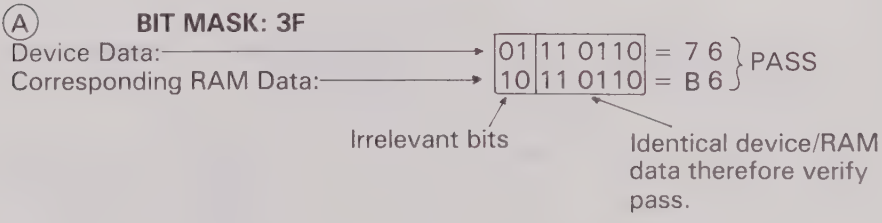
This is best shown by using an example for instance: 3F

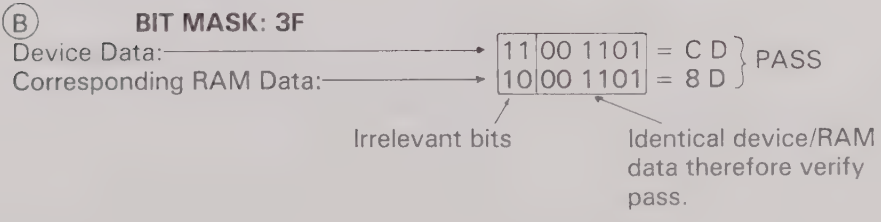

The Bit MASK is entered as a two digit hexadecimal figure:
3F
The PPZ converts this figure into its binary state:
0011 1111
The binary state can be translated into the key:
I I RR RRRR
I = Irrelevant bits
R = Relevant bits
Whereby 0's represent irrelevant bits and 1's represent relevant bits
3F effectively reduces the normal 8-Bit word to 6-Bits.

This means that the two most significant bits of the 8-Bit word can be either 0 or 1.

Therefore, the VERIFY TEST, with a BIT MASK of 3F will pass, so long as the remaining 6-bits are identical for all corresponding device/RAM locations within the specified address limits.

For example:



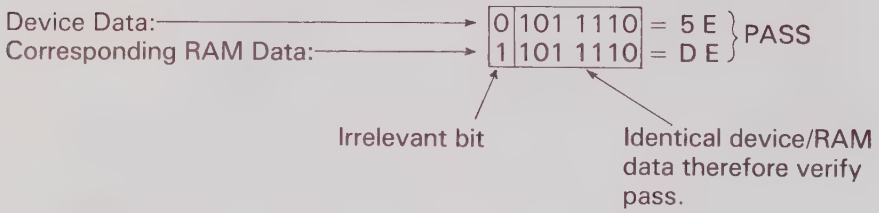


7F tends to be the most commonly required BIT MASK as it is associated with the ASCII code.

```
7F = 0111 1111 = IRRR RRRR = 7-BIT WORD

I = Irrelevant Bits
R = Relevant Bits
```

For example:

BIT MASK: 7F

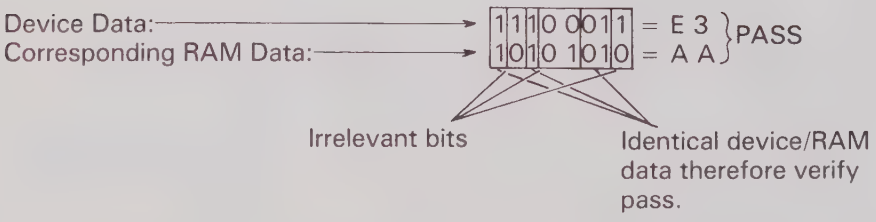


More unorthodox BIT MASKS can be entered if required for instance: A6

```
A6 = 1010 0110 = RIRI IRRI

I = Irrelevant bits
R = Relevant bits
```

For example:

BIT MASK: A6


Device Data:
Corresponding RAM Data:
1 1 1 0 0 0 1 1 = E 3
1 0 1 0 1 0 1 0 = A A
PASS
Irrelevant bits
Identical device/RAM data therefore verify pass.

The most extreme version of a BIT MASK is 00, whereby every data bit in an 8-Bit word becomes irrelevant. Therefore VERIFY will always pass whatever the differences are between the RAM data and the device data.

00 = 0000 0000 = IIII IIII I = Irrelevant bits

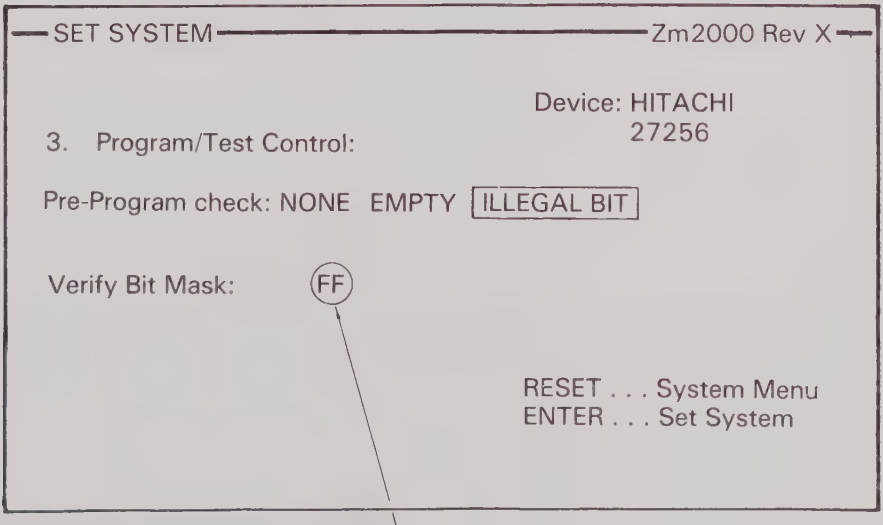
For example:
BIT MASK: 00

Device Data: 0101 1101 = 5 D
Corresponding RAM Data: 1010 0010 = A 2
} PASS

Irrelevant bits

ENTERING THE BIT MASK

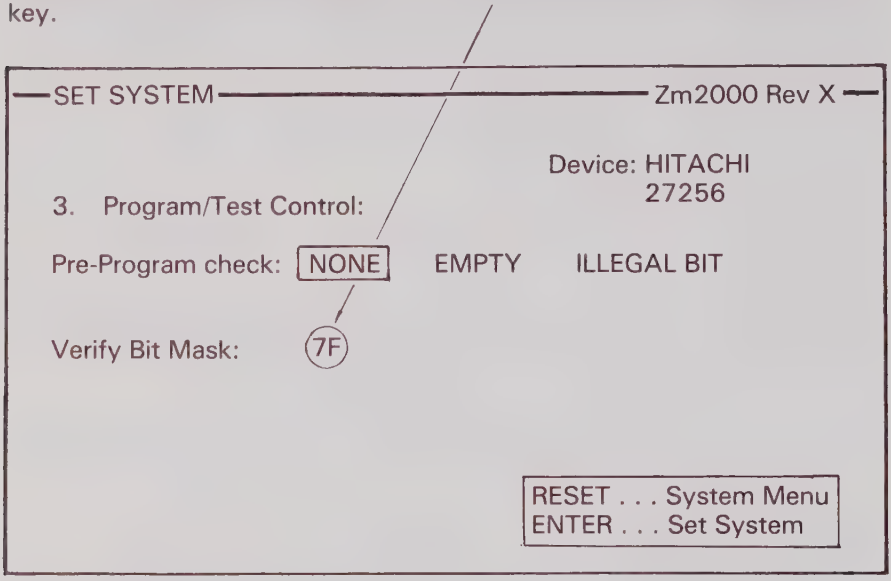
The selection can be made by first pressing RESET to return to the system menu and then pressing Key . . . A 'Set System' followed by Key . . . 3 'Program/Test Control' to display:



The bit mask will be in its default state FF.

'FF' means all 8 Bits of a word are relevant

Using the cursor keys will show FF flashing, the new bit mask can be selected by use of the hex-keyboard and entered by pressing the ENTER key.



The RESET key will return to the system menu, where the TEST mode can be re-entered.

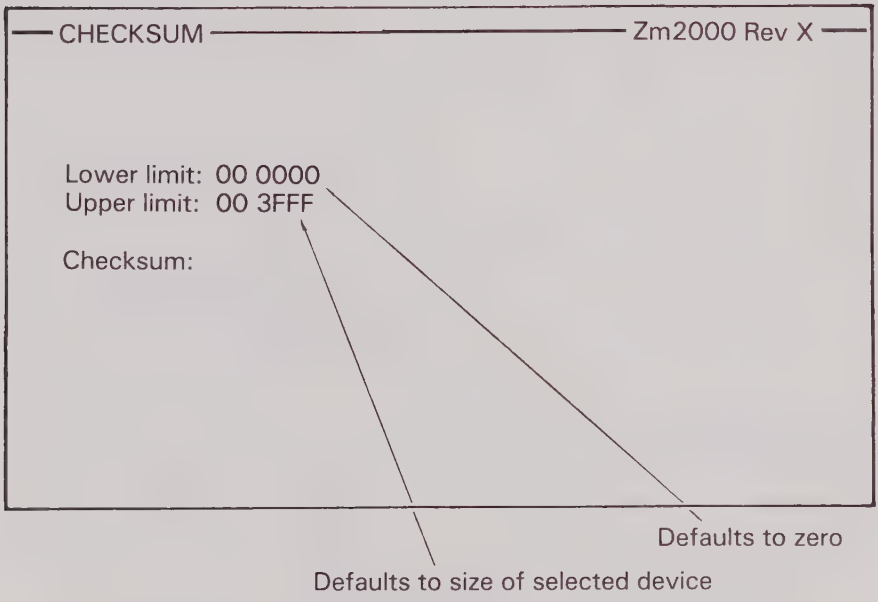
4.7 IN-PROGRAM: CHECKSUM

In-Program Checksum is an automatic addition of all byte size data programmed into the device.

MANUAL: CHECKSUM

The Manual Checksum of the RAM can be applied at any time. To apply the checksum, from the SYSTEM MENU:

PRESS KEY 8 to display:



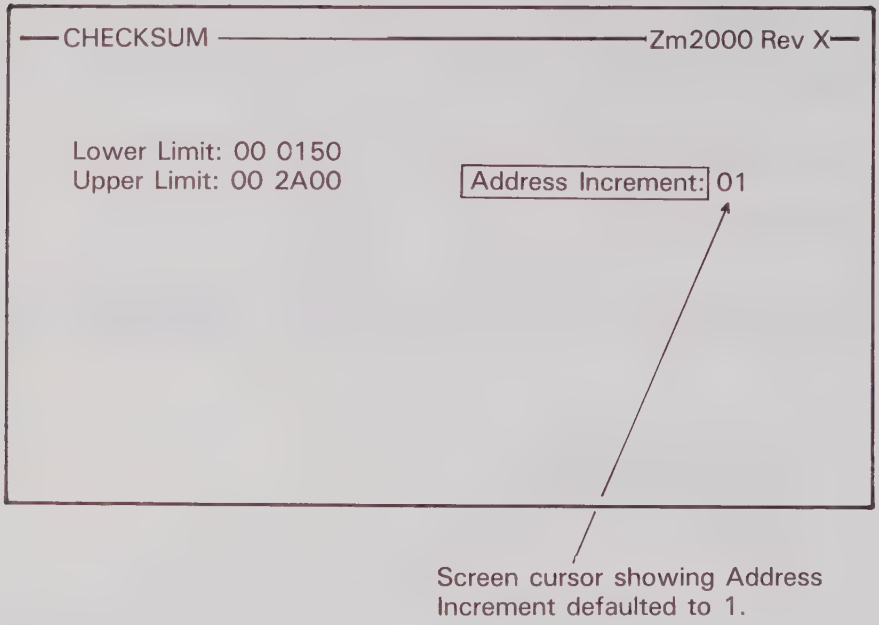
Address limits can be independently selected for the manual checksum (unlike the automatic versions which default to limits pre-selected in their respective operations i.e. Program, Load, Verify etc). Interchange between the two limits is made with the vertical cursor keys and new selections can be entered from the hex-keyboard.

NOTE: Only 4 digits can be entered if the memory extension is not used.

To confirm the RAM address limits:

PRESS ENTER

After having confirmed the address limits the screen will display:



If an Address Increment other than 1 is required it can now be entered from the keyboard.

To confirm the Address Increment and implement the Checksum:

PRESS ENTER

The checksum will be displayed in the bottom left corner of the screen.

To return the SYSTEM MENU:

PRESS RESET

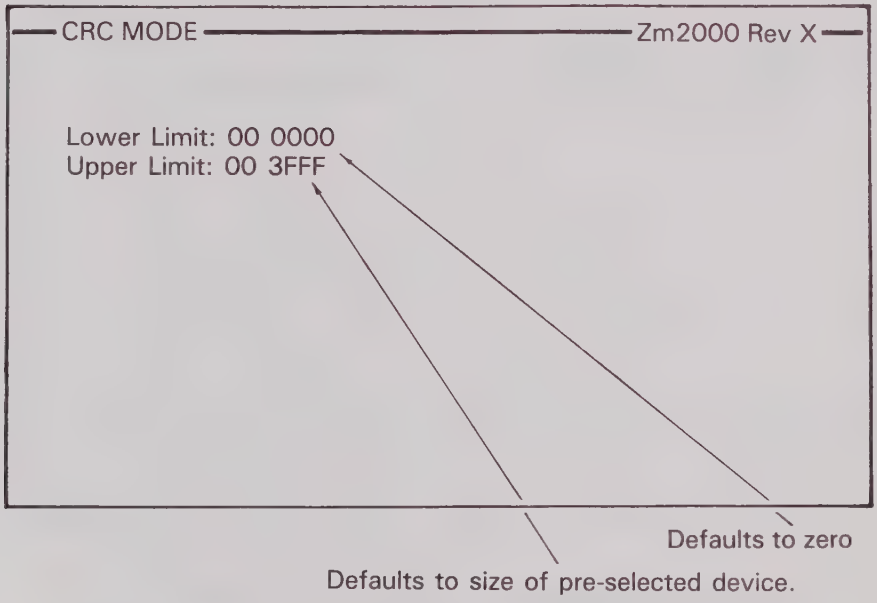
4.8 CYCLIC REDUNDANCY CHECK (CRC)

The Cyclic Redundancy Check applies a continuous process of shifting and addition to the RAM data. This yields a coded representation of the data, which is sensitive to the ordering of the data bytes unlike checksum which only considers their values.

To apply a CRC from the System Menu:

PRESS KEY 9

This will display:



Both Upper and Lower Address Limits can be changed, and also an Address Increment can be selected as with CHECKSUM.

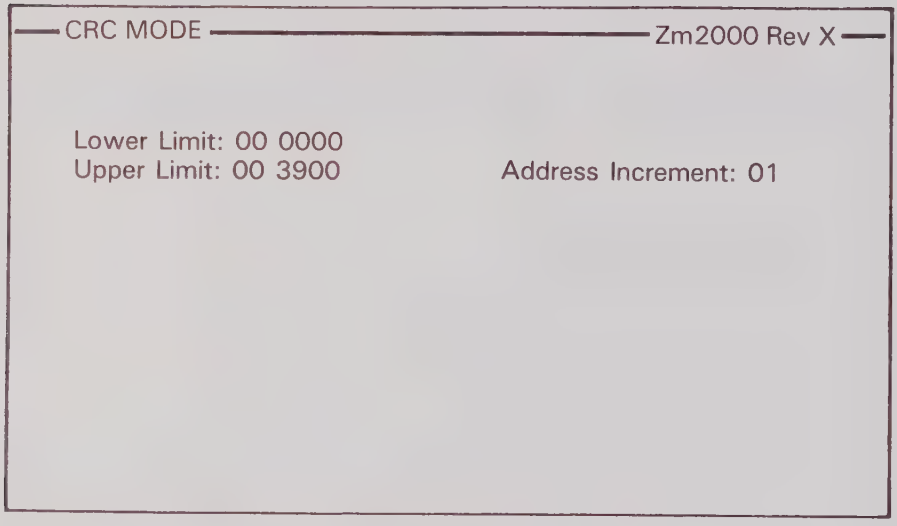
NOTE: Only 4 digits can be entered if the memory extension is not used.

CRC: Selecting Address Limits and Address Increment

As with CHECKSUM, interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓), and new selections can be entered directly from the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER to display:



If an Address Increment other than 1 is required it can now be entered from the keyboard.

To confirm the Address Increment and implement the CRC:

PRESS ENTER

When the check is complete, a four digit hex character indicating the result will appear on the screen.

For example:

Cyclic Redundancy Check: A768

To return to the SYSTEM MENU:

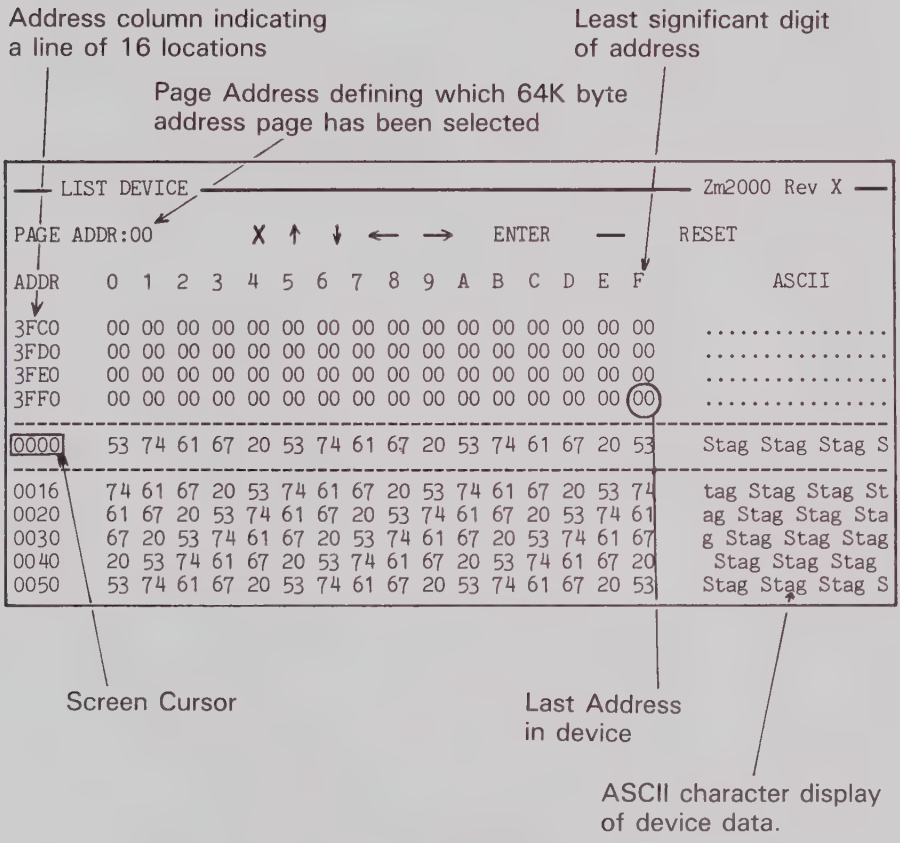
PRESS RESET

4.9 LIST DEVICE

This is a feature that allows the data content of the device to be scanned on the screen.

To list the device, from the SYSTEM MENU:

PRESS KEY 6 to display:



Scanning of Device Data

To scan the data:

PRESS KEY ↑ OR KEY ↓

The address will increment or decrement sixteen locations (one line) at a time. Alternatively an address can be entered from the hex-keyboard and upon pressing ENTER the specified area of device data will be displayed.

Since most devices only have a single page address, the horizontal keys (← →) are not operational.

'List device' only displays data in hexadecimal form, therefore pressing KEY X will not display octal or decimal forms.

SECTION 5

stag

Sophisticated systems for the discerning engineer.

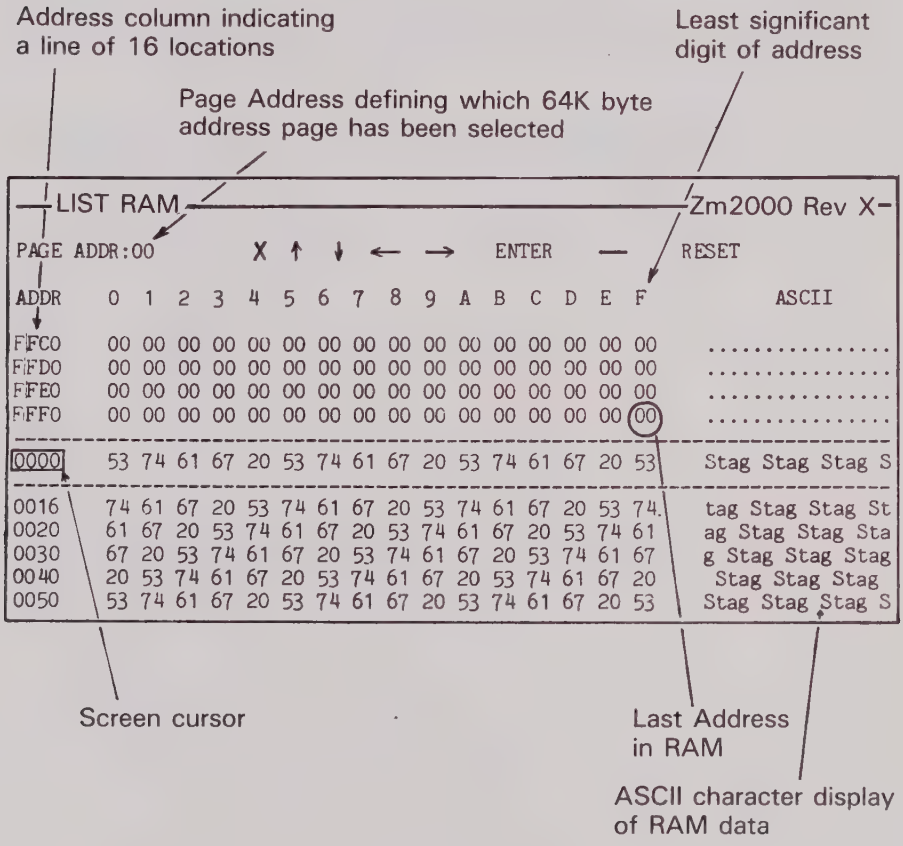
5.—RAM FUNCTIONS

5.1 LIST RAM

This is a feature which allows the data contents of the RAM to be scanned on the screen.

To LIST RAM, from the SYSTEM MENU:

PRESS KEY 5 to display:



Scanning of RAM Data

To scan the data:

PRESS KEY ↓ or KEY ↑

The address will increment or decrement sixteen locations (one line) at a time. Alternatively an address can be entered from the hex-keyboard and upon pressing ENTER the specified area of RAM data will be displayed.

If a RAM extension is fitted, then the horizontal cursor keys (← and →) or the keys H and L can be used to enable a page address other than 00 to be entered.

'List RAM' data is initially displayed in hexadecimal form. To display the data in decimal or octal forms:

PRESS KEY X (Once for decimal or twice for octal).

5.2 EDIT RAM

This is a feature whereby the actual content of the RAM can be directly modified by input from the keyboard.

There are nine EDIT functions available. To display them:

PRESS KEY 3

```
—EDIT RAM DATA ———————————— Zm2000 Rev X —

  1 . . . Hex Editor          6 . . . Nibble Swap
  2 . . . Decimal Editor      7 . . . String Search
  3 . . . Octal Editor        8 . . . Block Move
  4 . . . Logical OR Data     9 . . . Complement
  5 . . . Logical AND Data    RESET . . . System Menu
```

Entry into a particular EDIT function from the sub-menu (shown above) is made by pressing the appropriate key.

For instance to enter into the String Search function:

PRESS KEY 7

Note: For NIBBLE SWAP see section 3.5

5.2.1 HEX EDITOR

To enter the HEX EDITOR, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 1

The display will show a page of RAM data expressed in hexadecimal form on an address matrix 10 lines x 16 bytes in size. A screen cursor will show the default location 0000 which will be on the sixth line up between the two parallel dash lines running horizontally across the screen. At the right hand side of the screen there is an ASCII character representation of the data displayed in the adjacent matrix.

```
— HEX ENTER MODIFY DATA ———————————— Zm2000 Rev X —

Page ADDR: 00        X ↑ ↓ ←→    ENTER — RESET

ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    ASCII

FFC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
FFD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
FFE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
FFF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
-------------------------------------------------------------
[0000] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
-------------------------------------------------------------
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........
```

As with the LIST RAM data function the vertical cursor keys will increment or decrement the address by one line (sixteen locations) at a time. However, with the HEX EDITOR to transfer the screen cursor onto the data matrix:

PRESS ENTER or KEY →

In this way any location within the RAM can be found. When located, the data byte can be modified by the input of data from the hex-keyboard.

Alternatively (and faster) with the screen cursor in the default position 0000 an address can be entered from the keyboard and upon pressing ENTER the specified area of the RAM will be displayed and the screen cursor will indicate the chosen location.

To return the screen cursor to the address column:

PRESS KEY L

To display the RAM data in decimal or octal form:

PRESS KEY X (once for decimal or twice for octal).

To return the EDIT RAM DATA sub-menu:

PRESS KEY

5.2.2 DECIMAL EDITOR

To enter the DECIMAL EDITOR, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 2

The display will show a page of RAM data expressed in decimal form on an address matrix 10 lines x 10 columns in size. A screen cursor will show the default location 00000 which will be on the sixth line up between the two parallel dash lines running horizontally across the screen. At the right hand side of the screen there is an ASCII character representation of the data displayed in the adjacent matrix.

```
— DECIMAL ENTRY MODIFY DATA ————————— Zm2000 Rev X —

Page ADDR: 00        X ↑ ↓ ← →   ENTER — RESET

ADDR    0   1   2   3   4   5   6   7   8   9    ASCII

65500 000 000 000 000 000 000 000 000 000 000  ........
65510 000 000 000 000 000 000 000 000 000 000  ........
65520 000 000 000 000 000 000 000 000 000 000  ........
65530 000 000 000 000 000 000 000 000 000 000  ........
-------------------------------------------------------
[00000] 000 000 000 000 000 000 000 000 000 000  ........
-------------------------------------------------------
00010 000 000 000 000 000 000 000 000 000 000  ........
00020 000 000 000 000 000 000 000 000 000 000  ........
00030 000 000 000 000 000 000 000 000 000 000  ........
00040 000 000 000 000 000 000 000 000 000 000  ........
00050 000 000 000 000 000 000 000 000 000 000  ........
```

As with the LIST RAM data function the vertical cursor keys will increment the address by one line (in this case ten locations) at a time. However with the DECIMAL EDITOR to transfer the screen cursor onto the data matrix:

PRESS ENTER or KEY→

In this way any location within the RAM can be found. When located, the piece of data can be modified by the input of new data from the keyboard.

Alternatively (and faster) with the screen cursor in the default position 00000 an address can be entered from the keyboard and upon pressing ENTER the specified area of the RAM will be displayed and the screen cursor will indicate the chosen location.

To return the screen cursor to the address column:

PRESS KEY L

To display the RAM data in octal or hexadecimal form:

PRESS KEY X (once for octal or twice for hexadecimal)

To return the EDIT RAM DATA sub-menu:

PRESS KEY

5.2.3 OCTAL EDITOR

To enter the OCTAL EDITOR, FROM THE SYSTEM MENU:

PRESS KEY 3 followed by KEY 3

The display will show a page of RAM data expressed in octal form on an address matrix 10 lines x 8 columns in size. A screen cursor will show the default location 000000 which will be on the sixth line up between the two parallel dash lines running horizontally across the screen. At the right hand side of the screen there is an ASCII character representation of the data displayed in the adjacent matrix.

```
— OCTAL ENTER MODIFY DATA ———————————— Zm2000 Rev X —

 Page ADDR: 00        X ↑ ↓ ←→    ENTER — RESET

 ADDR     0   1   2   3   4   5   6   7     ASCII

 177740  000 000 000 000 000 000 000 000  ........
 177750  000 000 000 000 000 000 000 000  ........
 177760  000 000 000 000 000 000 000 000  ........
 177770  000 000 000 000 000 000 000 000  ........
 ------------------------------------------------
 000000  000 000 000 000 000 000 000 000  ........
 ------------------------------------------------
 000010  000 000 000 000 000 000 000 000  ........
 000020  000 000 000 000 000 000 000 000  ........
 000030  000 000 000 000 000 000 000 000  ........
 000040  000 000 000 000 000 000 000 000  ........
 000050  000 000 000 000 000 000 000 000  ........
```

As with the LIST RAM data function the vertical cursor keys will increment or decrement the address by one line (in this case eight locations) at a time. However, with the OCTAL EDITOR to transfer the screen cursor onto the data matrix:

PRESS ENTER or KEY →

In this way any location within the RAM can be found. When located, the piece of data can be modified by the input of new data from the keyboard.

Alternatively (and faster) with the screen cursor in the default position 000000 an address can be entered from the keyboard and upon pressing ENTER the specified area of RAM will be displayed and the screen cursor will indicate the chosen location.

To return the screen cursor to the address column.

PRESS KEY L

To display the RAM data in hexadecimal or decimal form:

PRESS KEY X (once for hexadecimal or twice for decimal)

To return the EDIT RAM DATA sub-menu

PRESS KEY –

5.2.4 Logical OR Data

The 'OR' function

This is a modification function where one of three 'OR' equations is applied to an individual bit within a RAM byte. The type of equation applied is dependent upon the pattern discrepancy of bits that constitute the entered 'OR MASK', compared to corresponding bits within the RAM bytes.

OR equations

NB: In logic equations the OR function is usually represented by a plus sign.

(i) Remain as one

If a RAM bit was 'one' it will stay as 'one' no matter what the corresponding OR MASK might be, ie:

RAM bit + corresponding OR MASK bit = Same RAM bit

1	+	1	=	1
1	+	0	=	1

(ii) Remain as zero

A RAM bit will remain as 'zero' only if the corresponding OR MASK bit is also a 'zero', ie:

RAM bit + corresponding OR MASK bit = Same RAM bit

0	+	0	=	0

(iii) Changed to one

A RAM bit can be changed to a 'one' from a 'zero' if the corresponding OR MASK bit is a 'one', ie:

RAM bit + corresponding OR MASK bit = Changed RAM bit

0	+	1	=	1

Therefore point (iii) alone is the only cause of any modification to existing RAM

Operation of the OR mask function

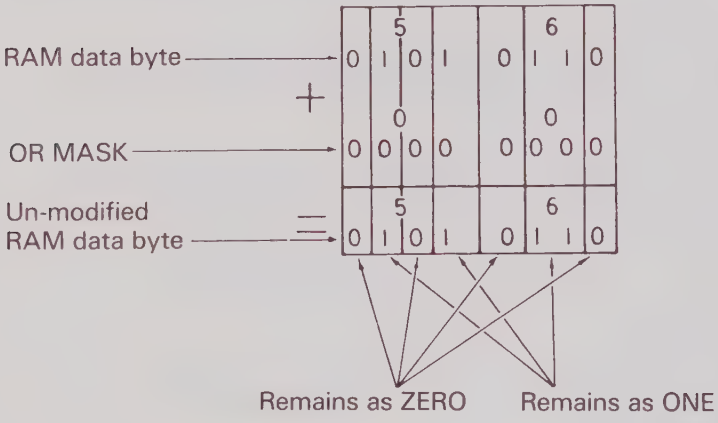
The operation of the OR MASK can be most clearly shown when the entered OR byte and the data bytes resident within the pre-selected RAM address limits are reduced to their fundamental binary form.

8 Bits = 1 Byte

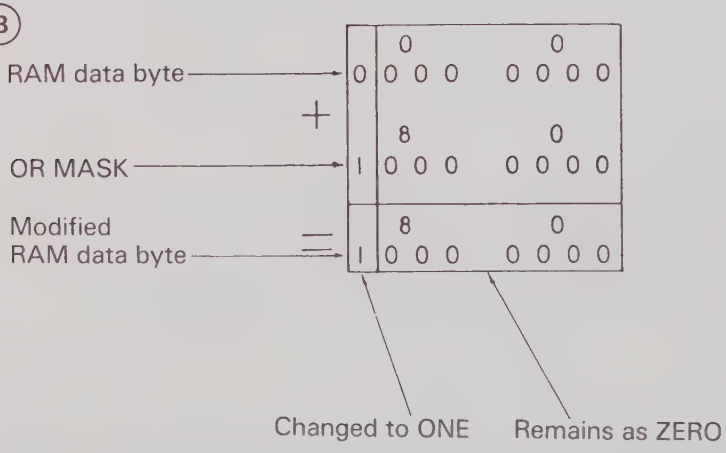
OR MASK examples:

(A) OR MASK 00

When '00' (the default condition of the OR MASK) is applied to any RAM data byte, it will remain unchanged.

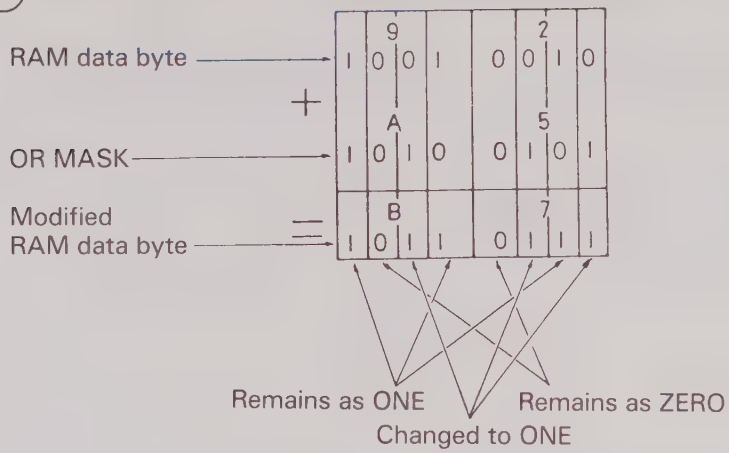


(B)

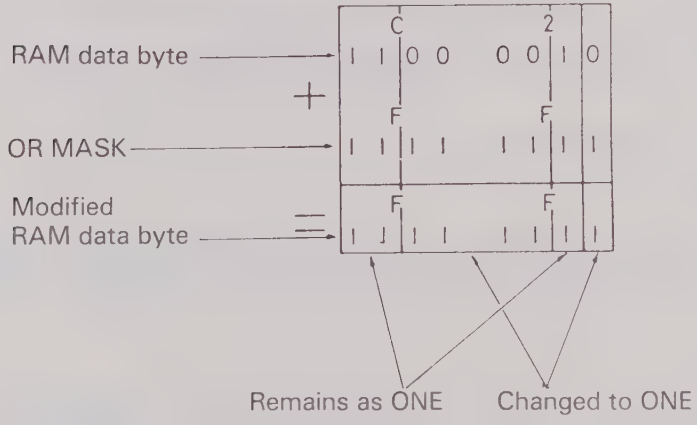


OR Mask examples continued

Ⓒ

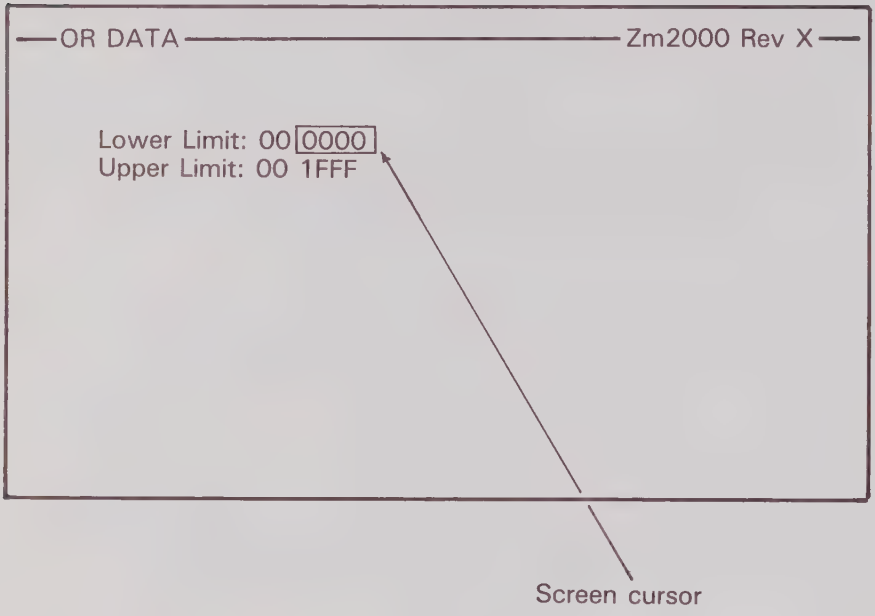


Ⓓ **OR MASK FF** When the OR MASK 'FF' is applied to any RAM data byte every bit will be modified to 'one'



To use the 'Logical OR Data' function, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 4 to display:



Interchange between the limits is made by using the vertical cursor keys and new values can be entered from the hex-keyboard.
Note: Only 4 digits can be entered if the memory extension is not used.

To confirm the address limits:

PRESS ENTER

'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08. To confirm the 'Address Increment':

PRESS ENTER

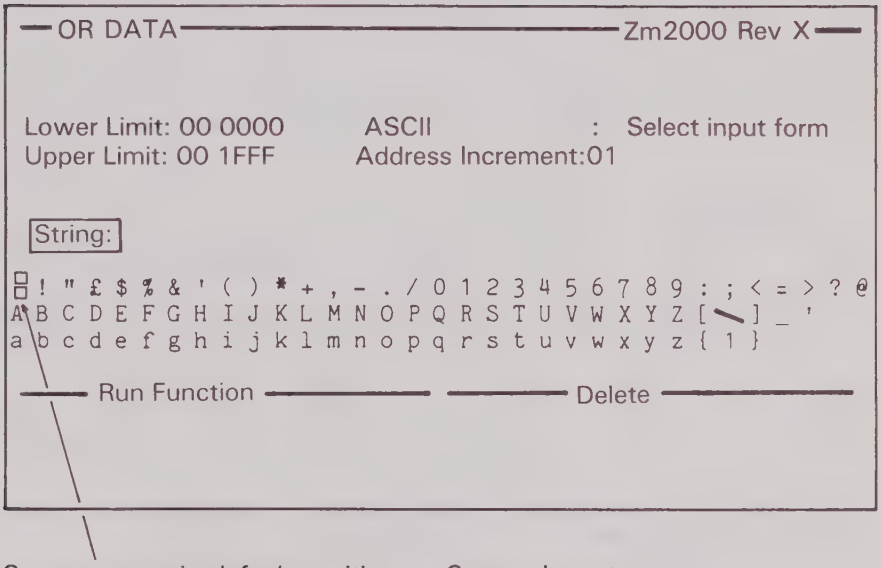
'Select input form' will start to flash beside the choice of ASCII and Numeric.

Selection of ASCII Form

PRESS KEY ←

The cursor will move to 'ASCII'

PRESS ENTER to display:


OR DATA
Zm2000 Rev X
Lower Limit: 00 0000
Upper Limit: 00 1FFF
ASCII : Select input form
Address Increment:01
String:
! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { | }
Run Function
Delete

Screen cursor in default position on Space character

To select a particular character, position the cursor over it by using all of the cursor keys and PRESS ENTER.

Characters will appear next to 'String' (up to 22) and may be deleted as required by moving the cursor onto the 'Delete' line and PRESSING ENTER.

To implement the function, move the cursor onto the 'Run Function' line and PRESS ENTER. This will also return the EDIT RAM DATA sub-menu.

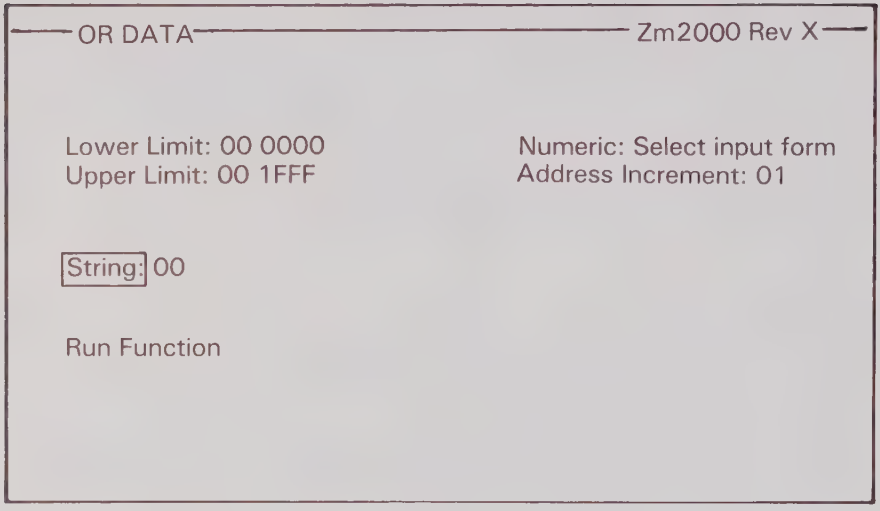
To return to the SYSTEM MENU:

PRESS RESET

Selection of Numeric Form

The default input form is 'Numeric':

PRESS ENTER to display:



The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

PRESS ENTER after each selected byte.

To delete a byte as required:

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER

This will return the EDIT RAM DATA sub-menu

To return to the SYSTEM MENU:

PRESS RESET

5.2.5 Logical AND Data

The 'AND' function

This is a modification function where one of three AND equations is applied to an individual bit within a RAM byte. The type of equation applied is dependent upon the pattern discrepancy of bits that constitute the entered AND MASK, compared to corresponding bits within the RAM bytes.

AND equations

NB: In logic equations the AND function is represented by a dot.

i. Remain as zero

If a RAM bit was 'zero' it will stay as 'zero' no matter what the corresponding AND MASK bit might be, ie:

RAM bit • corresponding AND MASK bit = Same RAM bit

0	•	0	=	0
0	•	1	=	0

ii. Remain as one

A RAM bit will remain as 'one' if the corresponding AND MASK bit is also a 'one' ie:

RAM BIT • corresponding AND MASK bit = Same RAM bit

1	•	1	=	1

iii. Changed to one

A RAM bit can be changed to 'zero' from a 'one' if the corresponding AND MASK bit is a 'zero', ie:

RAM bit • corresponding AND MASK bit = Changed RAM bit

1	•	0	=	0

Therefore point (iii) alone is the cause of any modification to existing RAM data bytes.

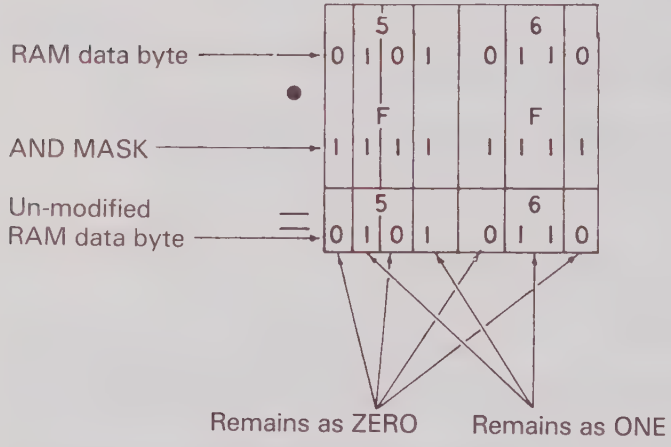
Operation of the AND mask function

The operation of the AND MASK can be most clearly shown when the entered AND byte and the data bytes resident within the pre-selected RAM address limits are reduced to their fundamental binary form.

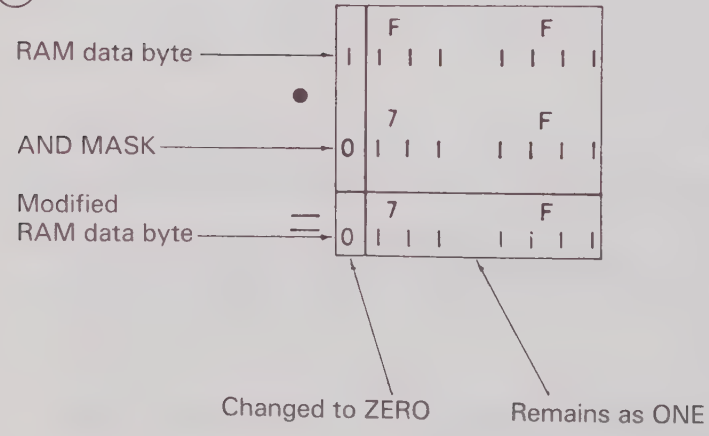
8 Bits = 1 Byte

AND MASK examples:

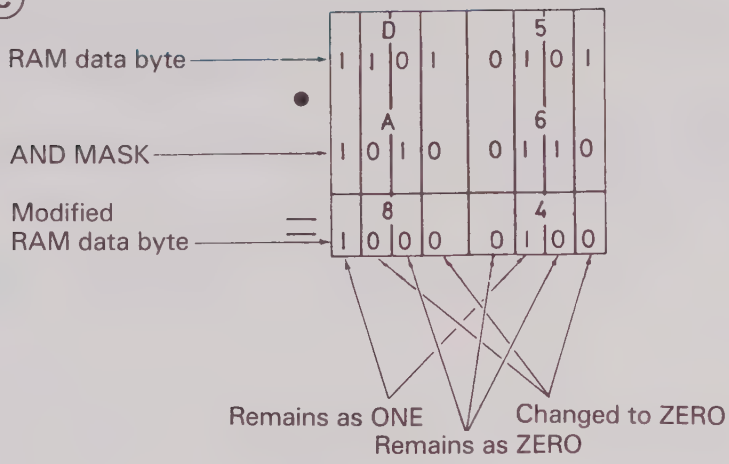
(A) **AND MASK FF** When FF (the default condition of the AND mask) is applied to any RAM data byte, it will remain unchanged.



(B)

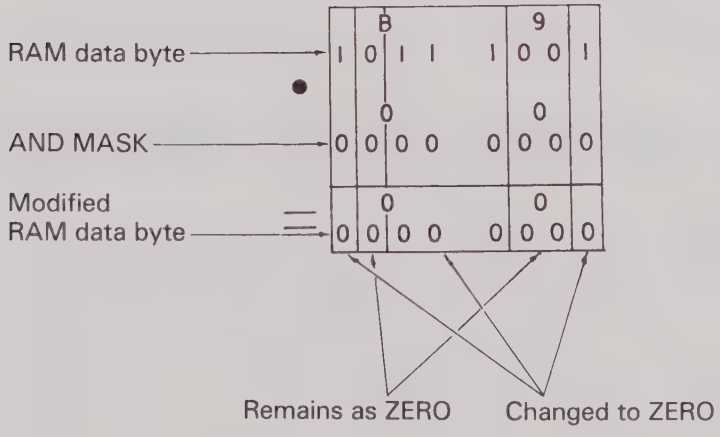


Ⓒ



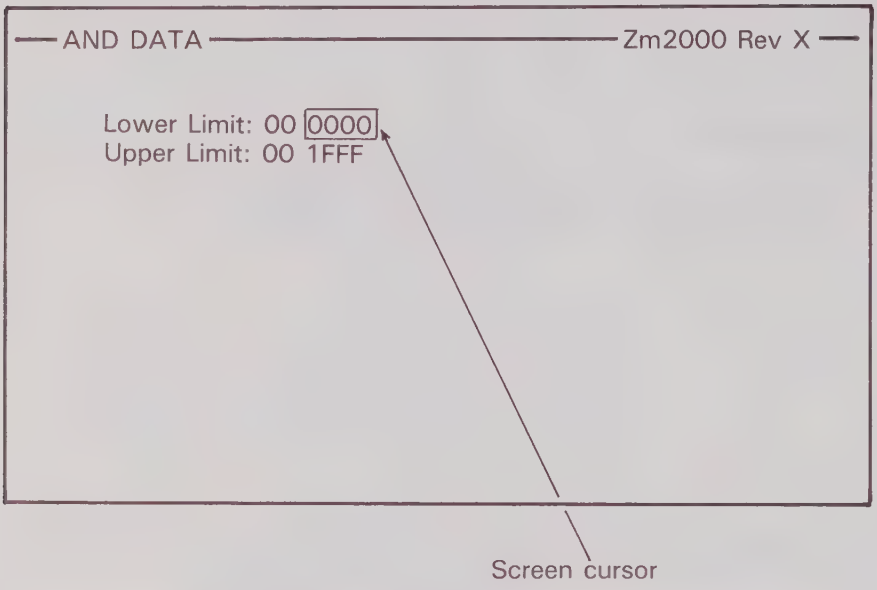
Ⓓ **AND MASK '00'**

When the AND MASK '00' is applied to any RAM data byte, every bit will be modified to zero.



To use the 'Logical AND Data' function, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 5 to display



Interchange between the limits is made by using the vertical cursor keys and new values can be entered from the hex-keyboard.
Note: Only 4 digits can be entered if the RAM extension is not used.

To confirm the address limits:

PRESS ENTER

'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08. To confirm the 'Address Increment':

PRESS ENTER

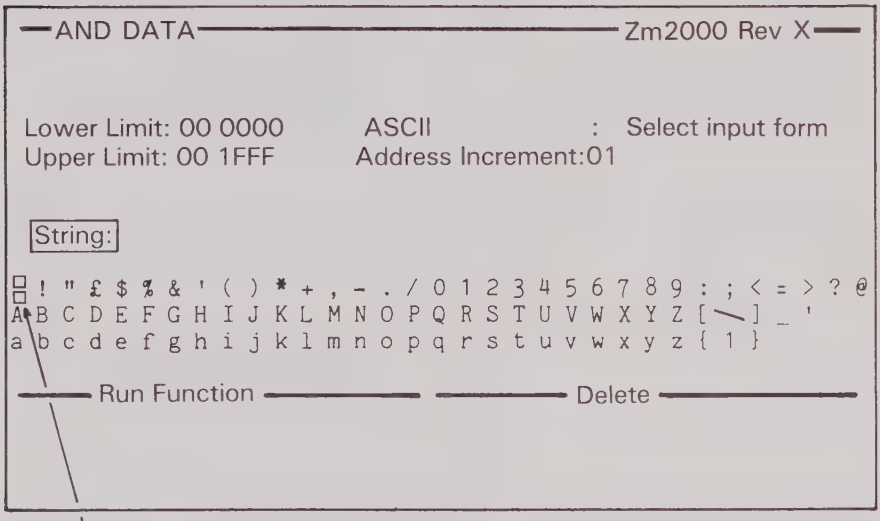
'Select input form' will start to flash beside the choice of ASCII and Numeric

Selection of ASCII form

PRESS KEY ←

The cursor will move to 'ASCII'

PRESS ENTER to display:


AND DATA
Zm2000 Rev X
Lower Limit: 00 0000
Upper Limit: 00 1FFF
ASCII
Address Increment:01
: Select input form
String:
! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { | }
Run Function
Delete

Screen cursor in default position on space character

To select a particular character, position the cursor over it by using all of the cursor keys and PRESS ENTER.

Characters will appear next to 'String' (up to 22) and may be deleted as required by moving the cursor onto the 'Delete' line and PRESSING ENTER.

To implement the function, move the cursor onto the 'Run Function' line and PRESS ENTER. This will also return the EDIT RAM DATA sub-menu.

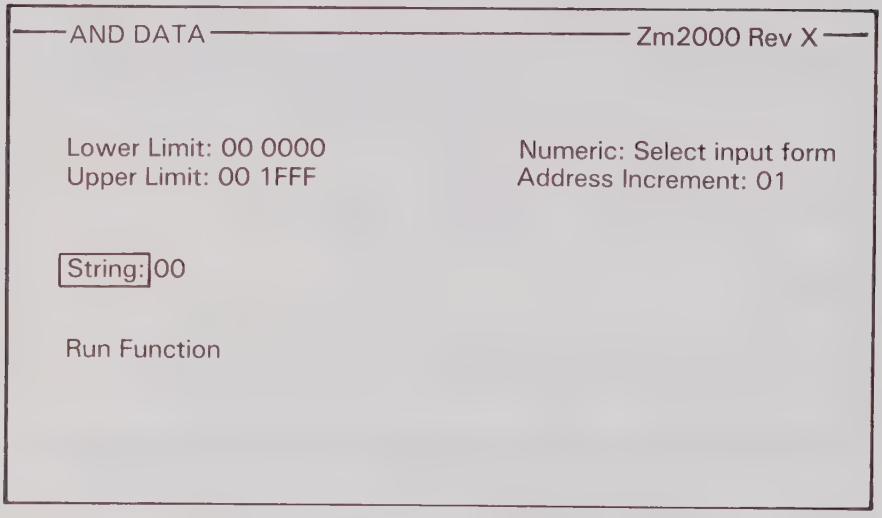
To return to the SYSTEM MENU:

PRESS RESET

Selection of Numeric form

The default input form is 'Numeric':

PRESS ENTER to display:



The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

PRESS ENTER after each selected byte.

To delete a byte (as required):

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER

This will return the EDIT RAM DATA sub-menu.

To return the system menu:

PRESS RESET

5.2.6 Nibble Swap—also see section 3.5

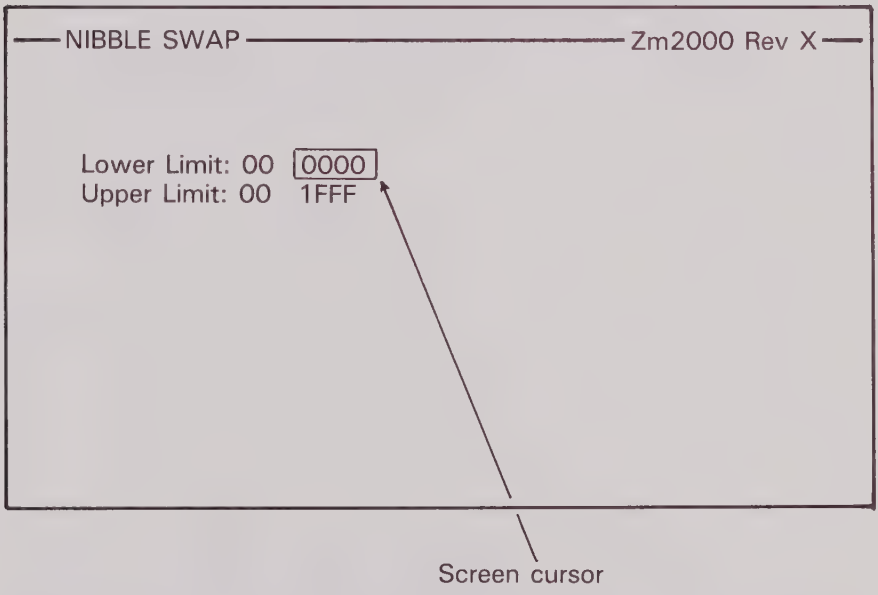
Two 4-bit devices can be programmed in two operations from a common address range within the RAM in such a way as to contain 8-Bit data when combined.

The address data will be 'split' during this programming sequence into two separate nibbles of data.

To enter the Nibble Swap mode:

PRESS KEY 3 followed by KEY 6.

The display will show the default state for the device:



Interchange between the limits is made by using the cursor keys and new values can be entered from the hex-keyboard. To confirm the address limits:

PRESS ENTER

After confirming the Address Limits, 'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08.

To confirm the address increment and implement the Nibble Swap:

PRESS ENTER

This will re-display the EDIT RAM DATA sub-menu.

To return the SYSTEM MENU:

PRESS ENTER

5.2.7 String Search

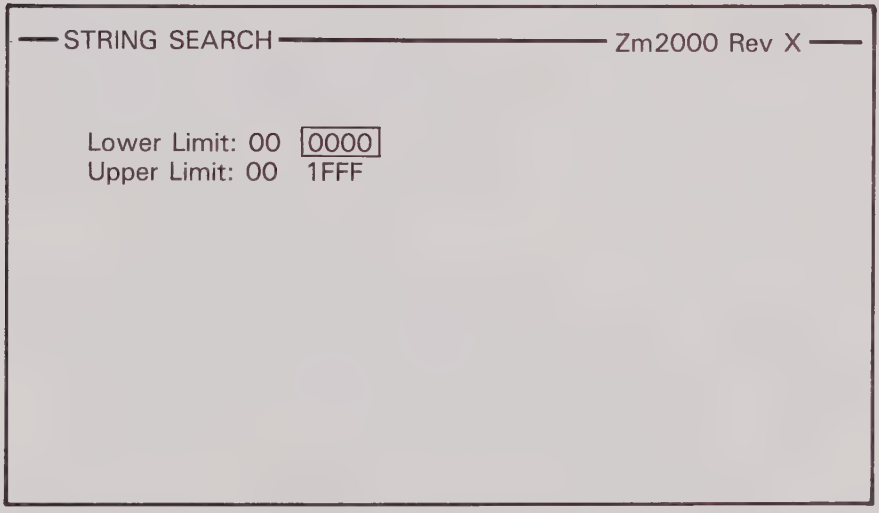
This is a feature which enables the user to search for a pattern of data in the RAM and to determine the address at which this data is stored.

A reference pattern can be input as a string of up to 22 hexadecimal data bytes or as a string of up to 22 characters from the ASCII register.

If a string is found, there is also a facility to display the page of RAM data within which the string is located.

To enter string search mode, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 7 to display:



Interchange between the RAM address limits can be made using the vertical cursor keys and new limits can be input from the hex-keyboard. Note: Only 4 digits can be entered if the RAM extension is not used.

To confirm the limits:

PRESS ENTER

'Select input form' will start to flash beside the choice of ASCII and Numeric.

Press ←

The cursor will move to 'ASCII'

Press ENTER

The display below appears on the screen:

```
──STRING SEARCH──────────────────────Zm2000 Rev X──

  Lower Limit: 00 0000      ASCII          :  Select input form
  Upper Limit: 00 1FFF      Address Increment:01

  String:

  □ ! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ '
  a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~

  ──── Run Function ────────  ──────── Delete ──────────
```

To select a particular character or function, position the cursor over it by use of the arrow keys and press ENTER.

Characters will appear next to 'string' and may be deleted as required using the 'Delete' function.

'Run Function' implements the ASCII string search.

If a matching string is present within the pre-selected RAM limits then the display will show the location of the first byte of the first string found:

```
──STRING SEARCH──────────────────────Zm2000 Rev X──

  Lower Limit: 00 0000      ASCII          :  Select input form
  Upper Limit: 00 1FFF

  String: $D
  → : Adjust specification   —: Return to sub-menu RESET: System Menu
  ENTER: Next occurence       E: View in the editor
  String found at: 00 0004
```

In this case the first match has been found at 00 0004.

To search for the next match:

PRESS ENTER

To view the match at its RAM location:

PRESS KEY E (to return to the current display PRESS —)

To adjust the specification for the string search:

PRESS KEY ↑

To return the RAM EDIT DATA sub-menu:

PRESS KEY

To return to the SYSTEM MENU:

PRESS RESET

ASCII String Search: Failure

If a matching string is not found within the pre-selected RAM limits, then the display will show:

```
——STRING SEARCH————————————————Zm2000 Rev X——

Lower Limit: 00 0000   ASCII        :  Select input form
Upper Limit: 00 1FFF

String: PROM
↑ : Adjust specification —: Return to sub-menu RESET: System Menu


String not found in specified limits
```

The up arrow key gives the chance to alter the limits for the search.

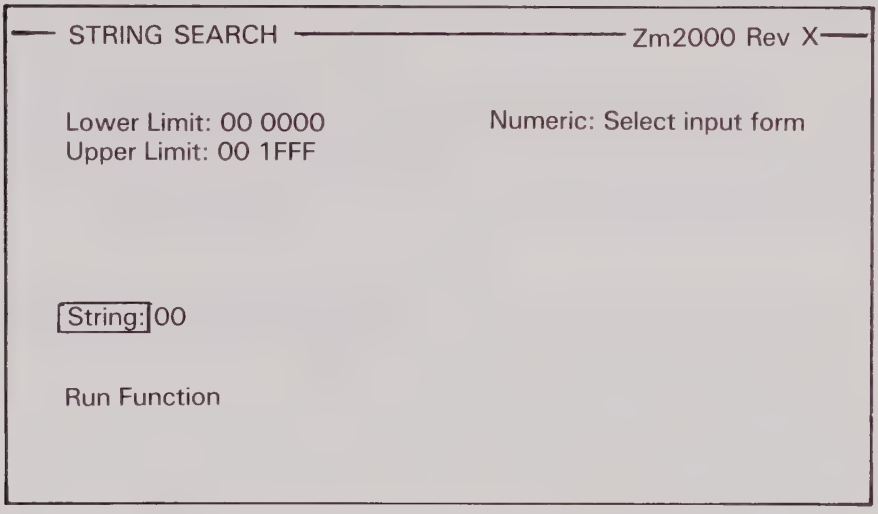
Press RESET

to return to the system menu.

Selection of Numeric form

The default input form is 'Numeric'.

PRESS ENTER to display:



The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

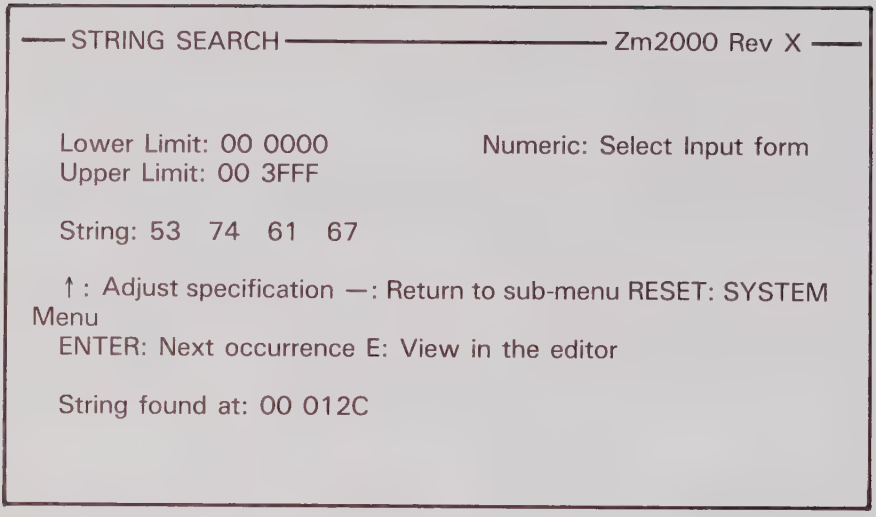
PRESS ENTER after each selected byte.

To delete a byte (as required):

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER


STRING SEARCH
Zm2000 Rev X
Lower Limit: 00 0000
Upper Limit: 00 3FFF
Numeric: Select Input form
String: 53 74 61 67
↑: Adjust specification —: Return to sub-menu RESET: SYSTEM Menu
ENTER: Next occurrence E: View in the editor
String found at: 00 012C

In this case the first match was found at 012C.

To search for the next match:

PRESS ENTER

To view the match at its RAM location:

PRESS KEY E (to return to the current display PRESS —)

To adjust the specification for the string search:

PRESS KEY ↑

To return the RAM EDIT DATA sub-menu:

PRESS KEY

To return to the SYSTEM MENU:

PRESS RESET

Numeric String Search: Failure

If a matching string is not found within the pre-selected RAM limits, then the display will show:

```
── STRING SEARCH ──────────────────── Zm2000 Rev X ──

Lower Limit: 00 0000   ASCII        Numeric: Select input form
Upper Limit: 00 1FFF

String: PROM
↑ : Adjust specification –:Return to sub-menu RESET: System Menu


String not found in specified limits
```

The up arrow key gives the chance to alter the limits for the search.

Press RESET

to return to the system menu.

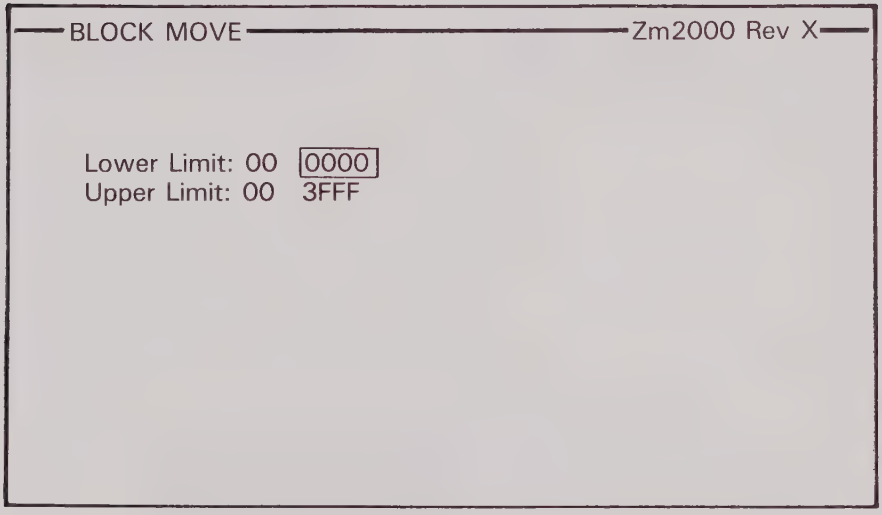
5.2.8 BLOCK MOVE

This is a feature enabling a block of data within pre-selected address limits to be re-located at another address within the RAM, without destroying the original data.

To enter BLOCK MOVE from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 8

This will display:



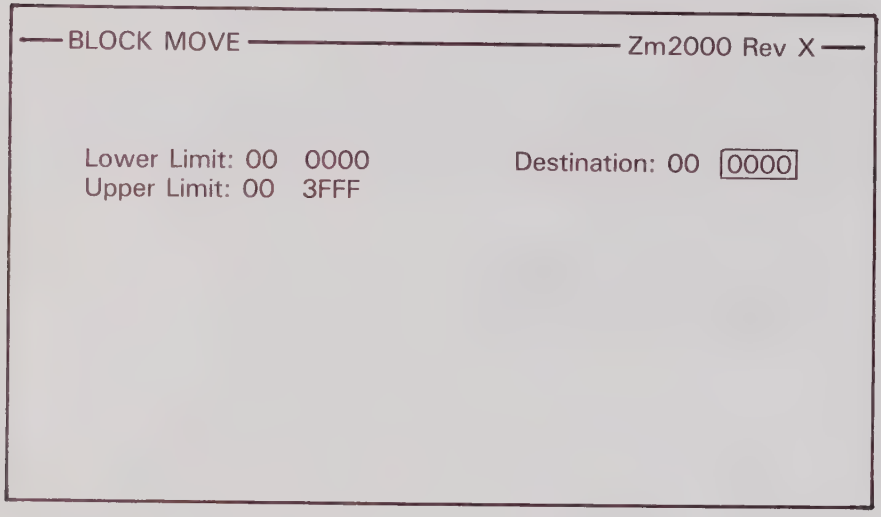
The display shows the upper and lower limits in their default states. The lower limit is defaulted to zero and the upper limit is defaulted to the size of the pre-selected device. These limits can be altered to define the size and location of the block of data to be moved.

Note: Only 4 digits can be entered if the RAM extension is not used.

Interchange between the two limits is made by use of the vertical cursor keys, and new values can be entered from the hex-keyboard. To confirm the new selections:

PRESS ENTER

The display will now show:



The screen cursor will now indicate that the lower limit of the destination to where the block of data is to be moved can be specified. This value can be entered from the hex-keyboard.

To confirm this value and implement the BLOCK MOVE:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

5.2.9 Complement

Complement is a modification of RAM data made by inverting the bits within the bytes.

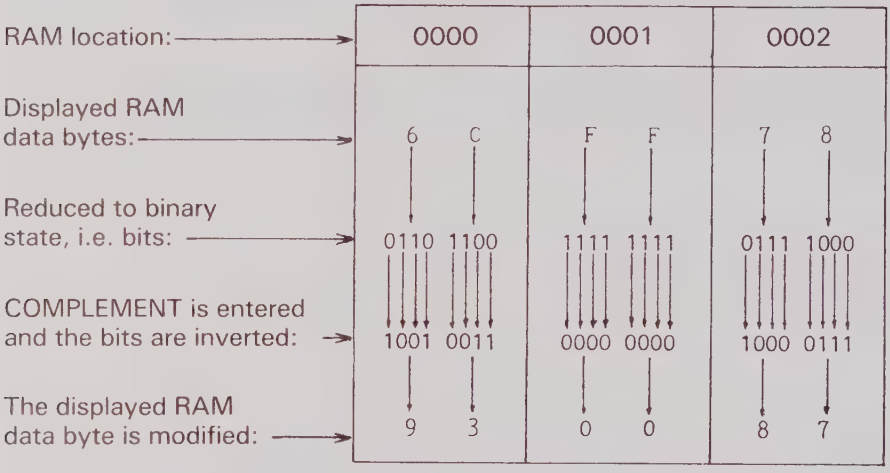
Complement will change 'zero' bits to 'ones' and 'one' bits to 'zeros' therefore transforming all RAM data bytes within pre-selected address limits.

COMPLEMENT OPERATION

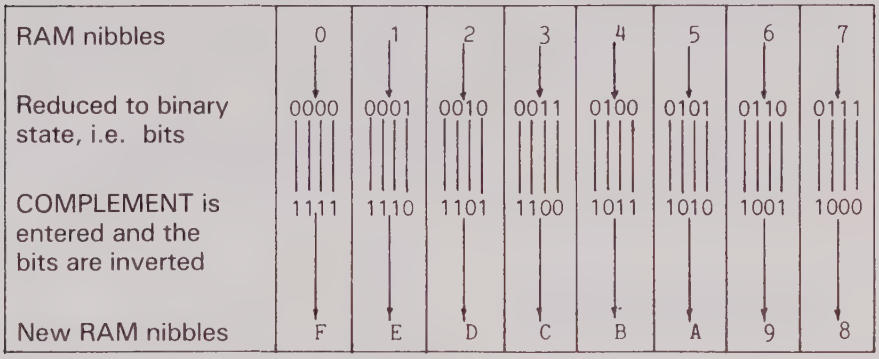
Complement is a modification of RAM bytes made by inverting the 'bits' that form the most fundamental constituent part.

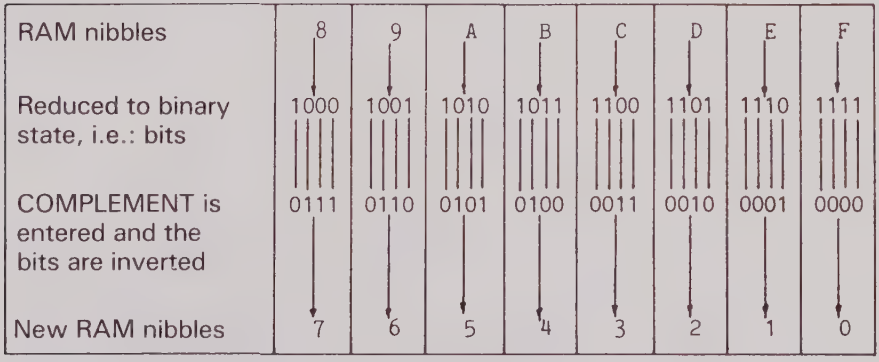
The COMPLEMENT will invert 'zero bits' to 'ones' and 'one bits' to 'zero', therefore transforming, all data bytes within pre-selected address limits.

For example:



A set pattern can be shown for the COMPLEMENT function by listing the 16 hexadecimal numerals as nibbles to undergo an inversion.

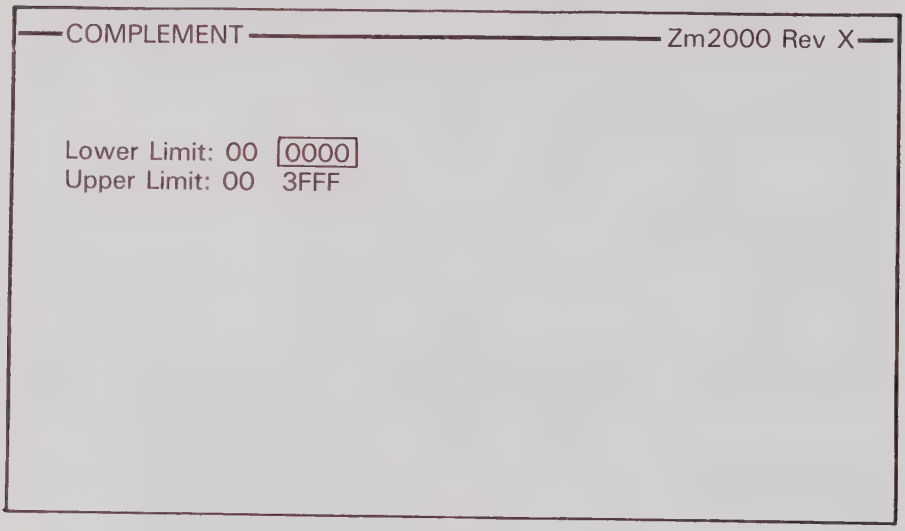

RAM nibbles
0 1 2 3 4 5 6 7
Reduced to binary state, i.e. bits
0000 0001 0010 0011 0100 0101 0110 0111
COMPLEMENT is entered and the bits are inverted
1111 1110 1101 1100 1011 1010 1001 1000
New RAM nibbles
F E D C B A 9 8


RAM nibbles
8 9 A B C D E F
Reduced to binary state, i.e.: bits
1000 1001 1010 1011 1100 1101 1110 1111
COMPLEMENT is entered and the bits are inverted
0111 0110 0101 0100 0011 0010 0001 0000
New RAM nibbles
7 6 5 4 3 2 1 0

Complement—Setting Address Limits

The COMPLEMENT function may be entered from the SYSTEM MENU by doing the following:

PRESS KEY 3 followed by KEY 9 to display:

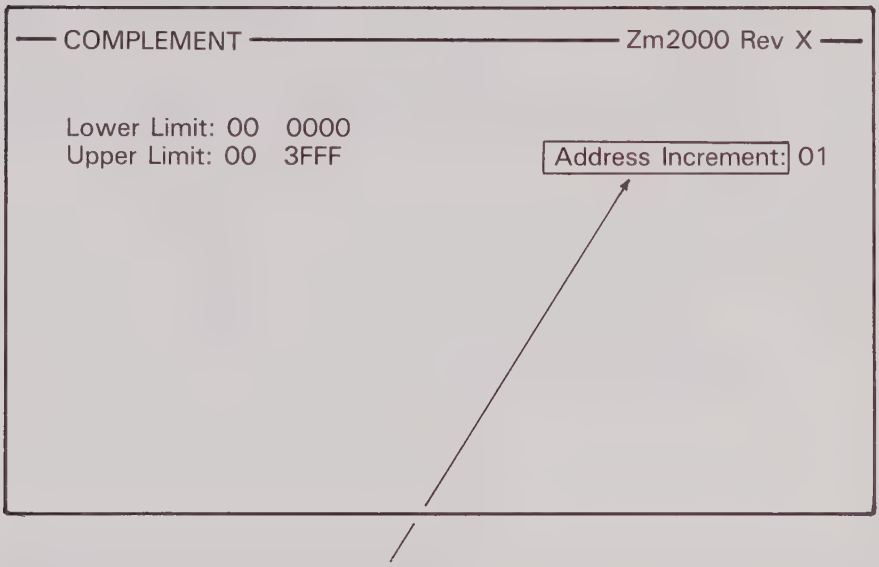


The address limits of the RAM to undergo the complement will be in their default states. The lower limit will be defaulted to zero and the upper limit to the size of the pre-selected device. Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and new selections can be entered from the hex-keyboard.

To confirm these selections:

PRESS ENTER

After the RAM address limits have been set, the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard, and can take values from 1 to 8.

To confirm the Address Increment and implement the Complement:

PRESS ENTER

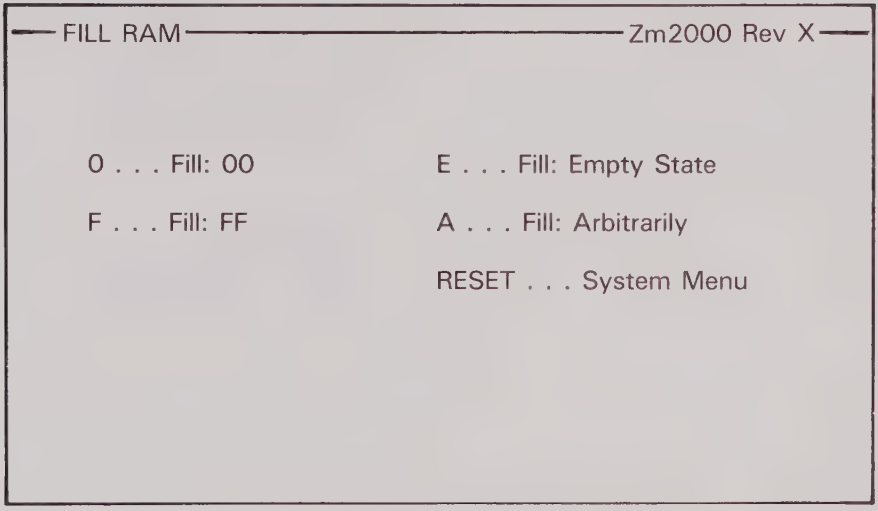
This will return the EDIT RAM DATA sub-menu. To return to the SYSTEM MENU:

PRESS RESET

5.3 FILL RAM

To enter the FILL RAM function from the SYSTEM MENU:

PRESS KEY 4 to display:



Four features are displayed, and entry into any particular one is made by pressing the appropriate key.

eg. to fill RAM with 00's

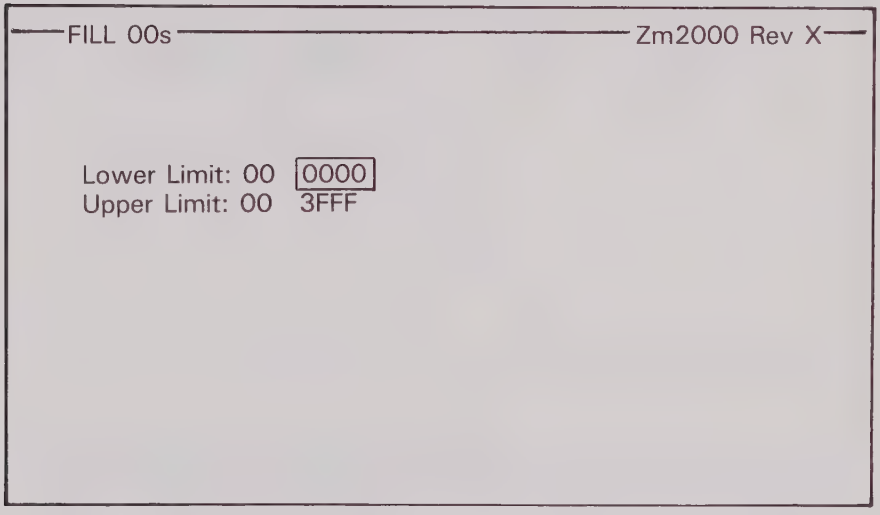
PRESS KEY 0

5.3.1 FILL RAM with 00—Setting Address Limits

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY 0

The display will show the default state of the RAM address limits to be filled with 00.



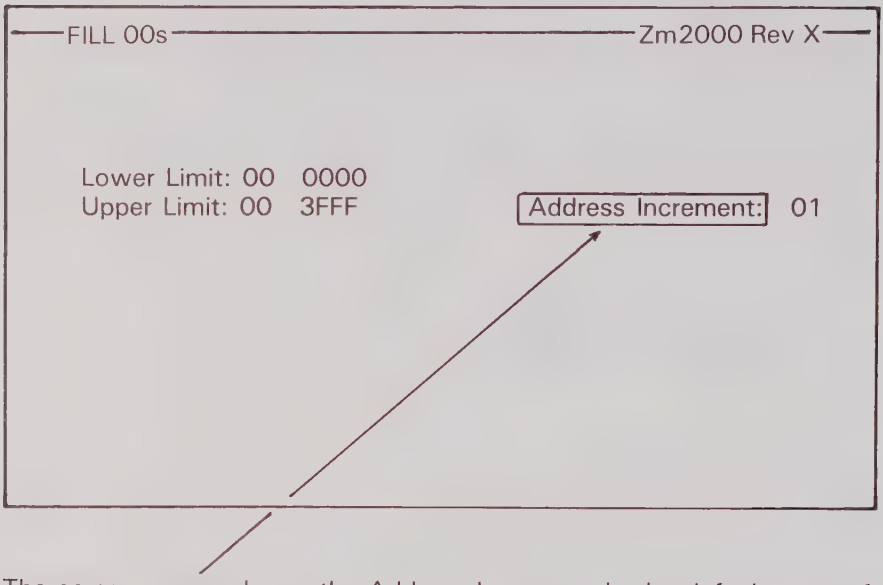
This lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment and fill the selected area of RAM with OOs:

PRESS ENTER

This will return the FILL RAM sub-menu. To return the SYSTEM MENU:

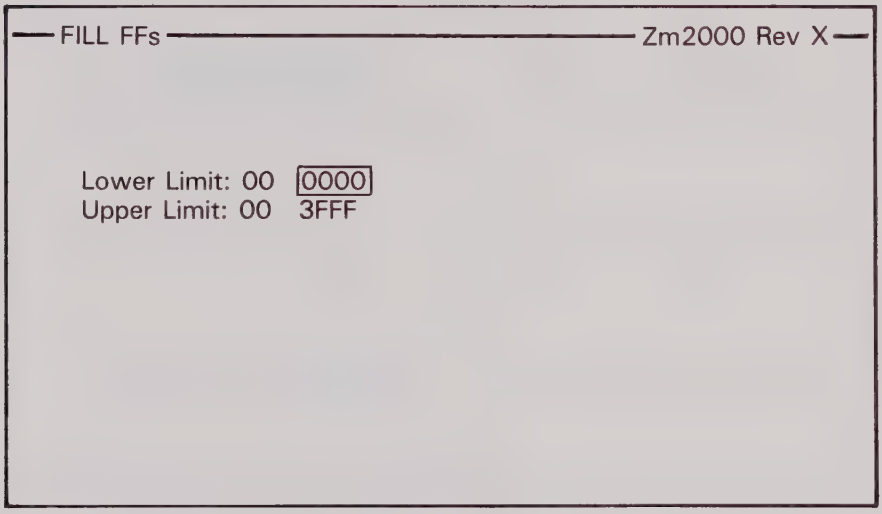
PRESS RESET

5.3.2 FILL RAM with FF—Setting Address Limits

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY F

The display will show the default state of the RAM address limits to be filled with FF.



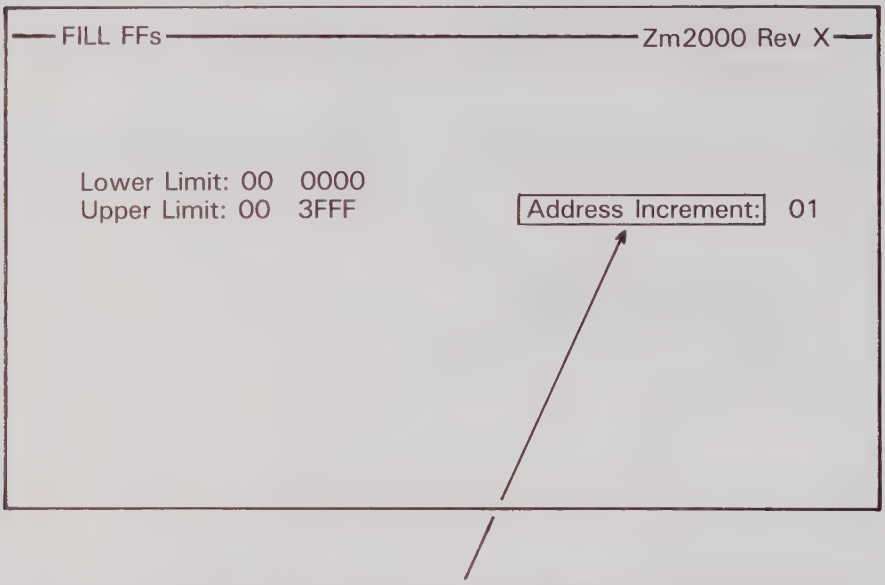
The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment and fill the selected area of RAM with FFs:

PRESS ENTER

This will return the FILL RAM sub-menu. To return the SYSTEM MENU:

PRESS RESET

5.3.3 FILL RAM with EMPTY STATE

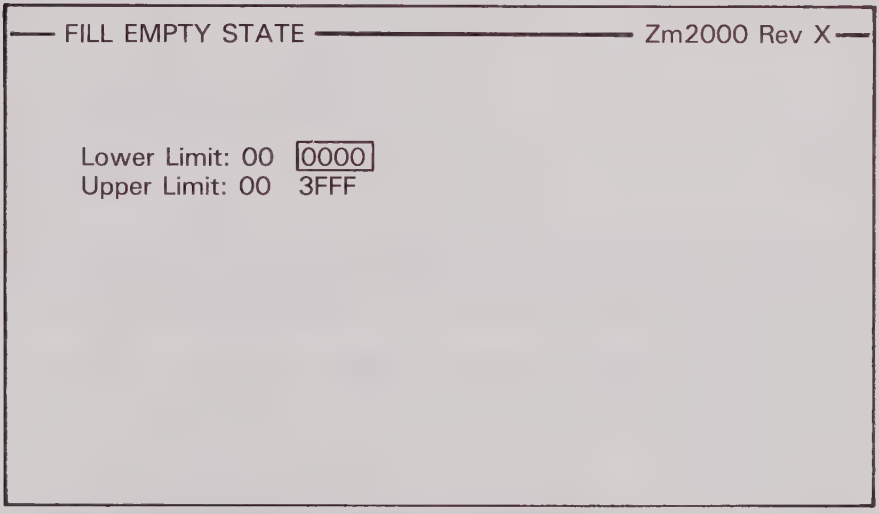
This feature duplicates the empty conditions of the pre-selected device and enters it into the specified area of RAM.

The EMPTY or UNPROGRAMMED STATE can vary for each device: 8-bit devices can be either FF or OO and 4-bit devices can be either F or 0.

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY E.

The display will show the default state of the RAM address limits to be filled with the 'EMPTY STATE'.



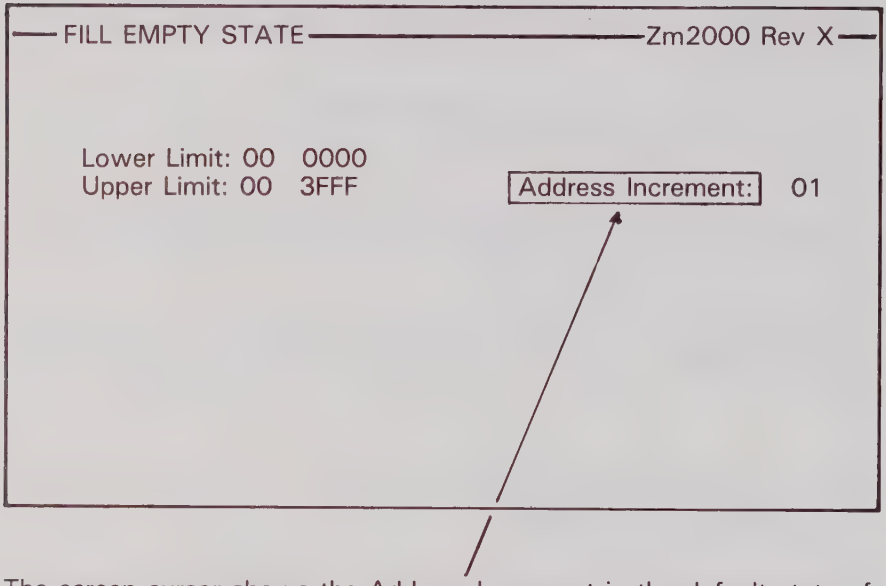
The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment and fill the selected area of RAM with 'EMPTY STATE'.

PRESS ENTER

This will return the FILL RAM sub-menu. To return the SYSTEM MENU:

PRESS RESET

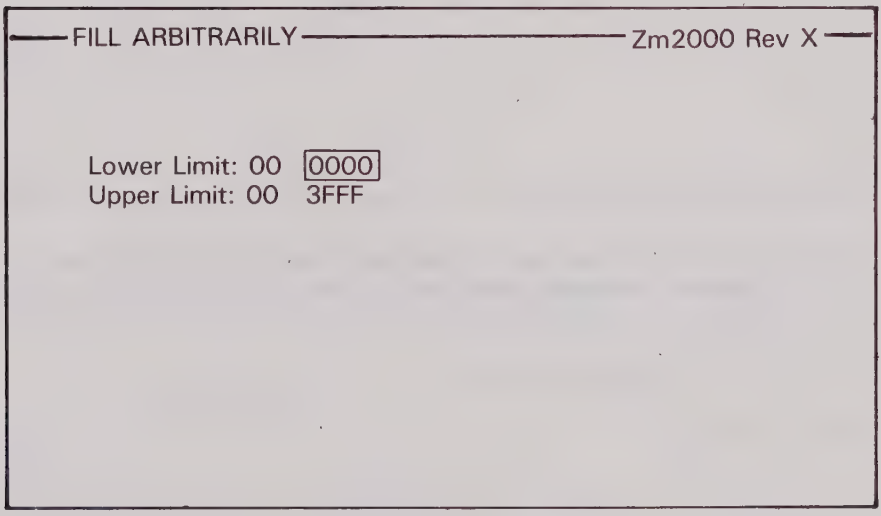
5.3.4 FILL RAM ARBITRARILY—Numeric (Hex Data)

This feature allows a string of up to 22 bytes of keyboard selected hex data to be entered in such a way as to be repeated throughout the entire specified area of RAM.

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY A

The display will show the default state of the RAM address limits to be filled with Hex Data.



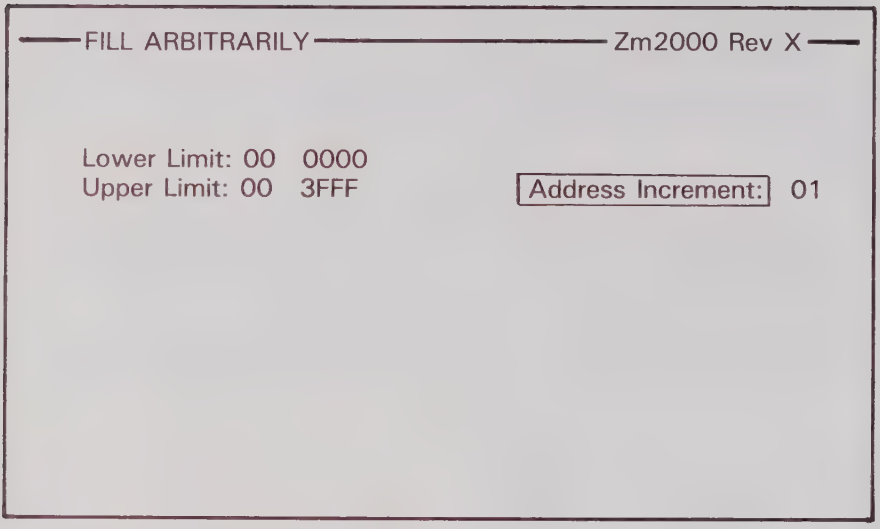
The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

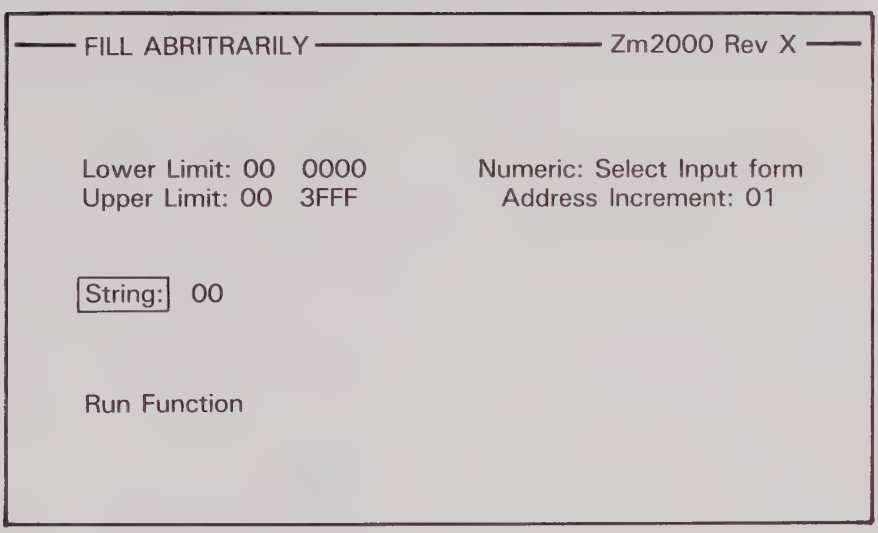
To confirm the Address Increment:

PRESS ENTER

To select the Numeric input form (hex data):

PRESS ENTER

The screen will now display:



A data string in bytes can be selected by use of the hex-keyboard. As each byte is selected, pressing ENTER will confirm the selection and allow the next byte to be chosen. Up to 22 bytes of hex data can be entered.

To delete bytes of data:

PRESS KEY←

This will delete the last selected byte.

When the desired string of data has been selected, to fill the specified area of RAM:

PRESS KEY ↓ followed by ENTER

This will return the FILL RAM sub-menu. To return to the SYSTEM MENU:

PRESS RESET

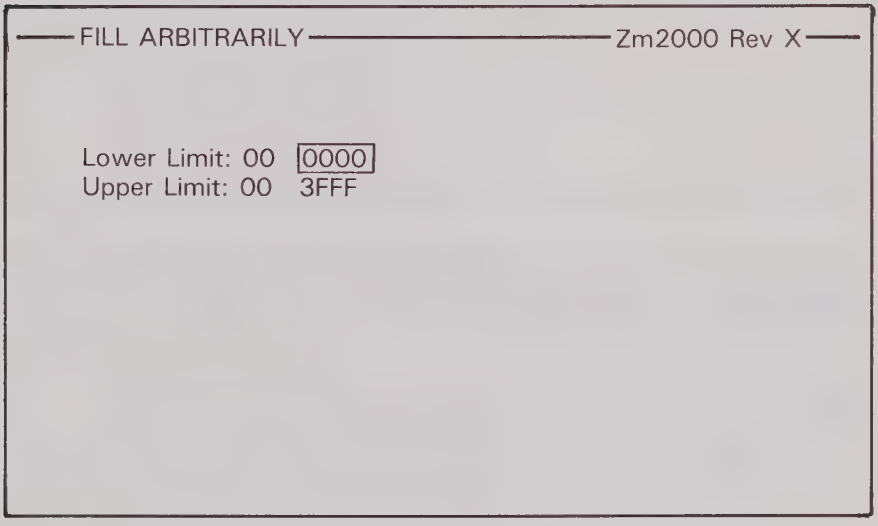
FILL RAM ARBITRARILY – ASCII

This feature allows a string of up to 22 characters from the ASCII register to be entered in such a way as to be repeated throughout the entire specified area of RAM.

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY A.

The display will show the default state of the RAM address limits to be filled with ASCII string data.



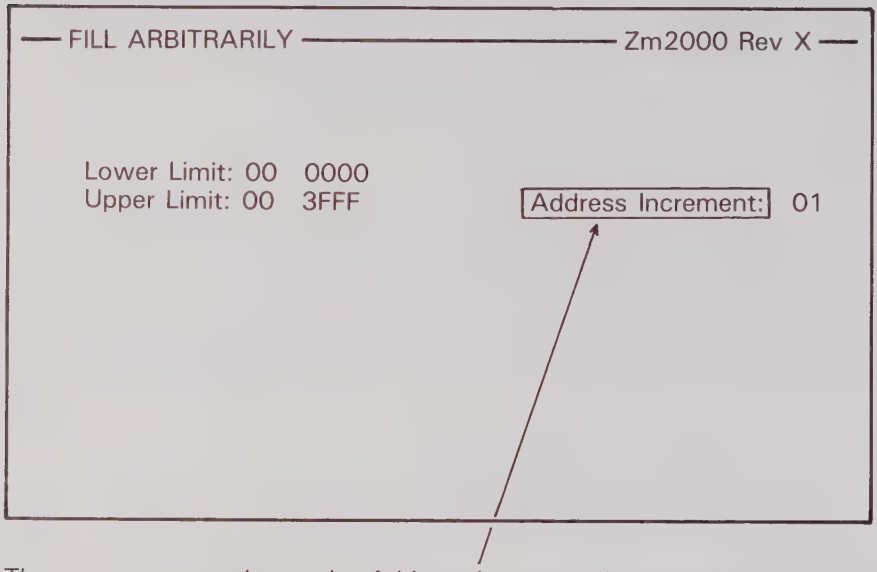
The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:



The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

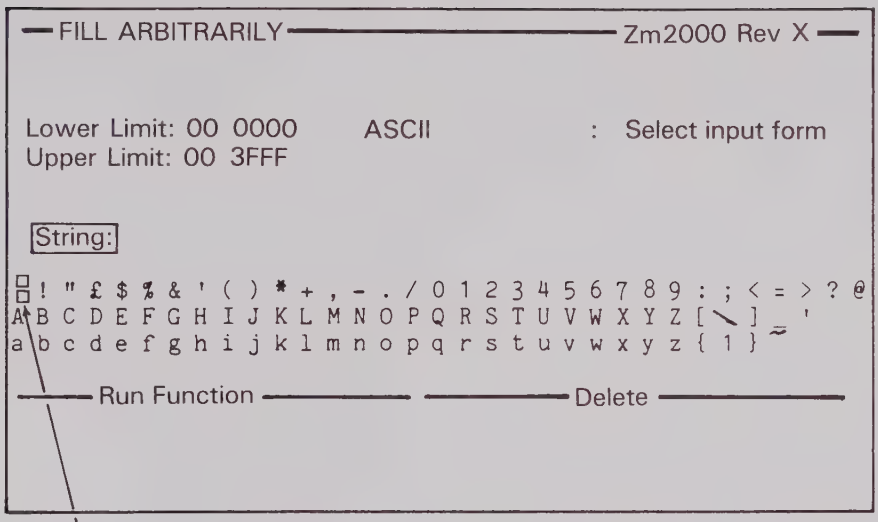
To confirm the Address Increment:

PRESS ENTER

To select the ASCII input form:

PRESS KEY ← followed by ENTER.

The screen will now display:


FILL ARBITRARILY
Zm2000 Rev X
Lower Limit: 00 0000
Upper Limit: 00 3FFF
ASCII
: Select input form
String:
! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { 1 }
Run Function
Delete

The screen cursor will now be in the default position on the space character. By using all of the cursor keys, any character from within the ASCII register can be pin pointed and then selected by pressing the ENTER key.

When the required string is displayed (up to 22 characters) moving the cursor down onto the 'Run Function' line and pressing ENTER will fill the specified RAM area with the ASCII string data, and return the FILL RAM sub-menu.

To return to the SYSTEM MENU:

PRESS RESET

SECTION 6

stag

Sophisticated systems for the discerning engineer.

6.—INTERFACE

6.1 SETTING THE I/O INTERFACE PARAMETERS (for both RS232C ports and the IEEE)

On power-up the I/O parameters default to the last selection.

There are five RS232C categories of pre-set I/O interface parameter available on the Zm2000 and one parameter that applies to the IEEE only.

They are divided under two separate headings: ''Set Interface Hardware'' and ''Set I/O Software'' which are displayed on two separate screen pages:

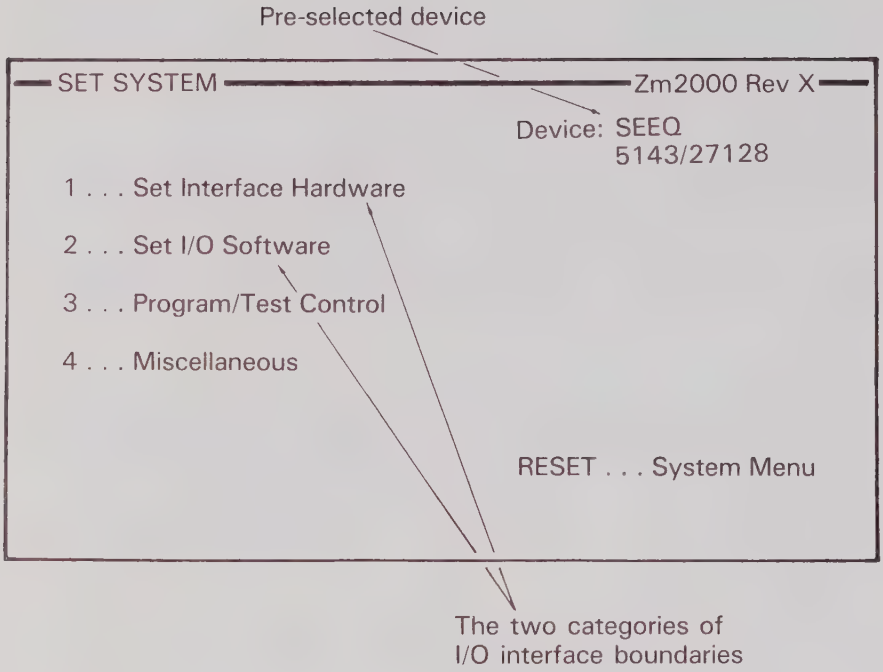
Interface parameters on Zm2000

Set Interface Hardware (Page 1)	Set I/O Software (Page 2)
Baud Rate Word Length Parity Stop Bits	Format Selection (plus control Z option)
IEEE Address Parameter	

Entry into both of these categories is first made by doing the following. From the SYSTEM MENU:

PRESS KEY A

This will display the SET SYSTEM sub-menu.


Pre-selected device
SET SYSTEM
Zm2000 Rev X
Device: SEEQ
5143/27128
1 . . . Set Interface Hardware
2 . . . Set I/O Software
3 . . . Program/Test Control
4 . . . Miscellaneous
RESET . . . System Menu
The two categories of
I/O interface boundaries

From the SET SYSTEM sub-menu

To enter "Set Interface Hardware":

PRESS KEY 1

To enter "Set I/O Software":

PRESS KEY 2

SET INTERFACE HARDWARE

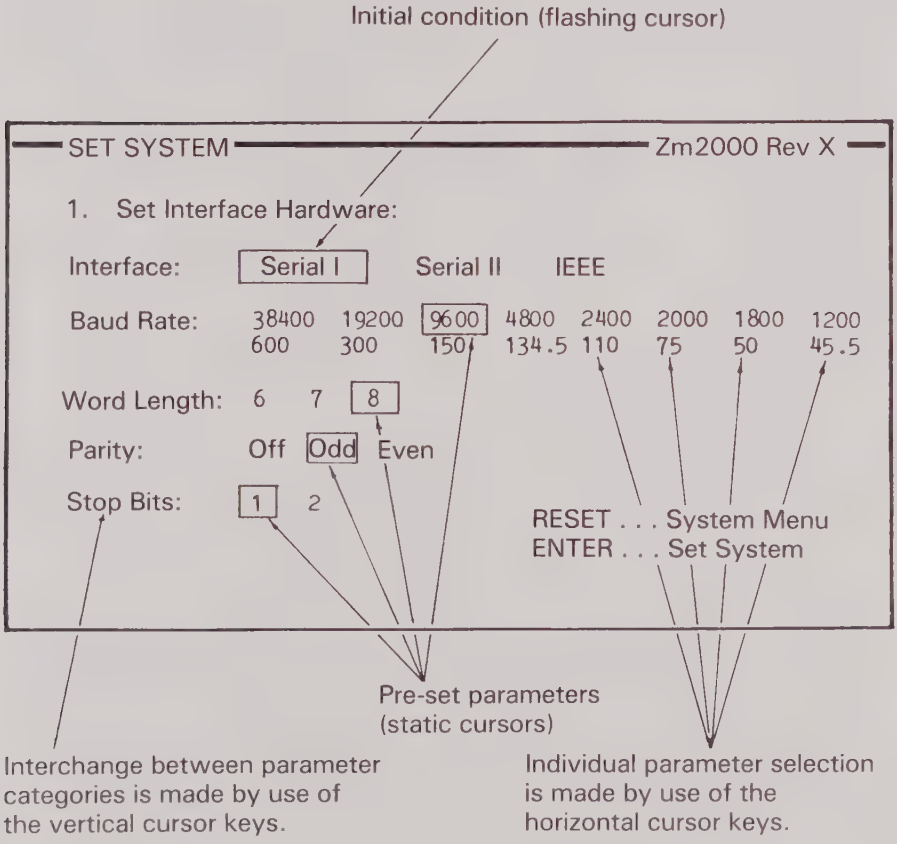
Hardware parameters for each of the three interface ports can be specified independently.

Selecting Hardware parameters for the Serial I port (RS232C female connector)

From the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the following:


Initial condition (flashing cursor)
SET SYSTEM
Zm2000 Rev X
1. Set Interface Hardware:
Interface: Serial I Serial II IEEE
Baud Rate: 38400 19200 9600 4800 2400 2000 1800 1200
600 300 150 134.5 110 75 50 45.5
Word Length: 6 7 8
Parity: Off Odd Even
Stop Bits: 1 2
RESET . . . System Menu
ENTER . . . Set System
Pre-set parameters
(static cursors)
Interchange between parameter categories is made by use of the vertical cursor keys.
Individual parameter selection is made by use of the horizontal cursor keys.

The flashing screen cursor will be initially on the heading Serial I (the RS232C female connector), and four static cursors will indicate the previously entered I/O parameters for this particular port.

Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the latest entered selection to the required alternative.

For example: To change the Baud Rate from 9600 as displayed in the diagram to 2400.

PRESS KEY ↓ followed by:

PRESS KEY → twice

The screen cursor will now be flashing on the new Baud Rate selection of 2400.

After all new parameters have been specified,

PRESS ENTER

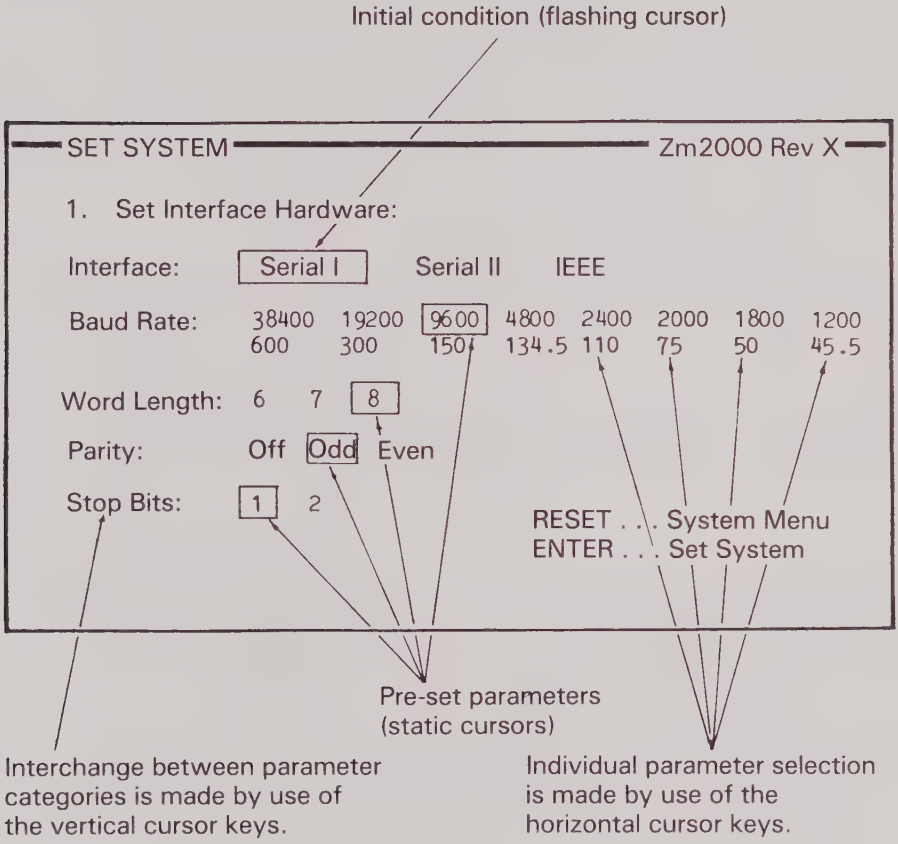
This will return the SET SYSTEM sub-menu. To return to the SYSTEM MENU:

PRESS RESET

From the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the following:

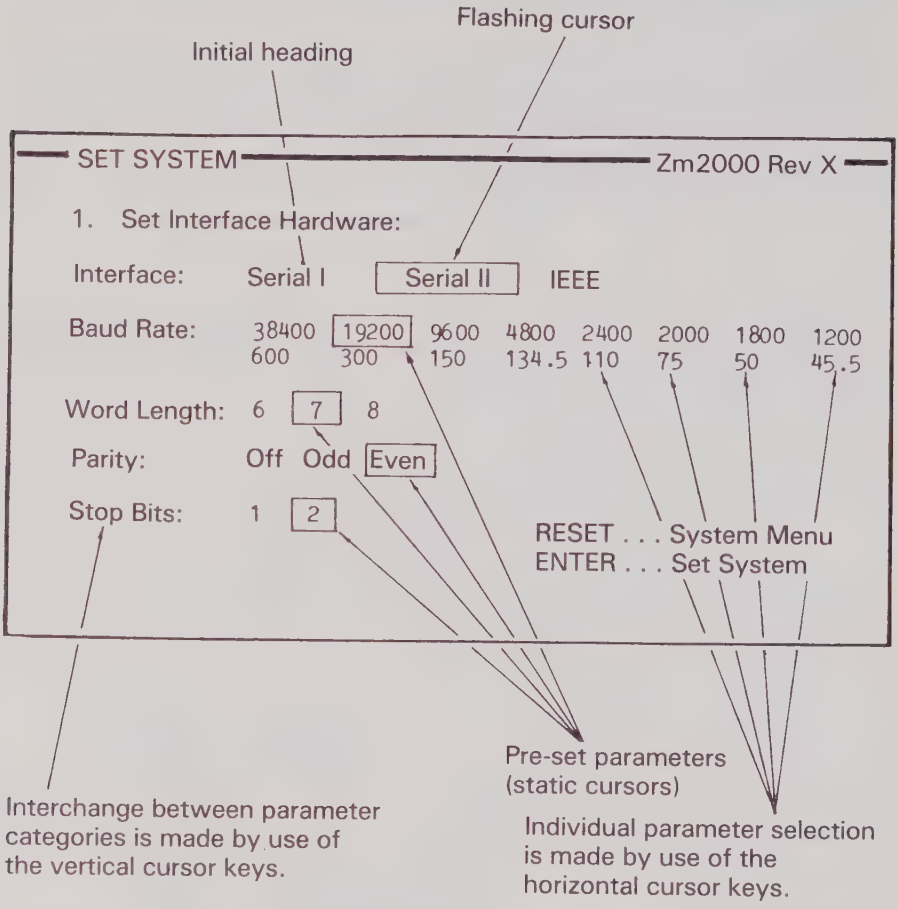

Initial condition (flashing cursor)
SET SYSTEM
Zm2000 Rev X
1. Set Interface Hardware:
Interface: Serial I Serial II IEEE
Baud Rate: 38400 19200 9600 4800 2400 2000 1800 1200
600 300 150 134.5 110 75 50 45.5
Word Length: 6 7 8
Parity: Off Odd Even
Stop Bits: 1 2
RESET . . . System Menu
ENTER . . . Set System
Pre-set parameters
(static cursors)
Interchange between parameter categories is made by use of the vertical cursor keys.
Individual parameter selection is made by use of the horizontal cursor keys.

The flashing screen cursor will be in its default state on the Serial I heading. To select the Serial II category.

PRESS KEY →

With the selection of the Serial II port, all the parameters indicated will be those previously selected for this configuration.

After Serial II has been selected, the screen will display:


Flashing cursor
Initial heading
SET SYSTEM
Zm2000 Rev X
1. Set Interface Hardware:
Interface: Serial I Serial II IEEE
Baud Rate: 38400 19200 9600 4800 2400 2000 1800 1200
600 300 150 134.5 110 75 50 45.5
Word Length: 6 7 8
Parity: Off Odd Even
Stop Bits: 1 2
RESET . . . System Menu
ENTER . . . Set System
Pre-set parameters
(static cursors)
Interchange between parameter categories is made by use of the vertical cursor keys.
Individual parameter selection is made by use of the horizontal cursor keys.

Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

To confirm the selections:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting the address parameter for the IEEE (IEEE-488 Talker/Listener Interface Port

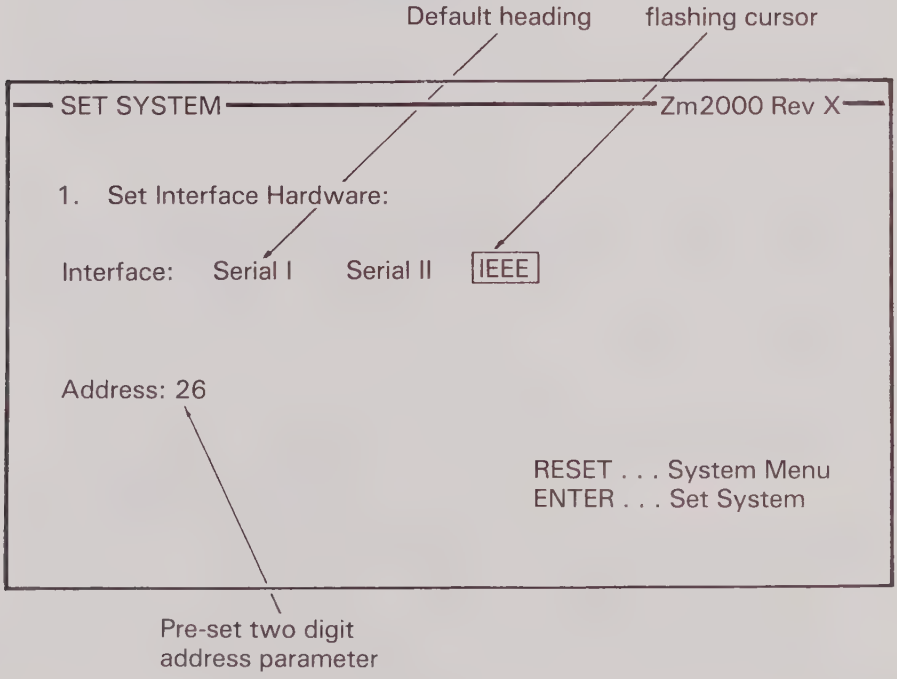
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the PPZ Programmer.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the Serial I heading. To select the IEEE:

PRESS KEY → twice to display:



Pressing the down cursor key (↓) will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

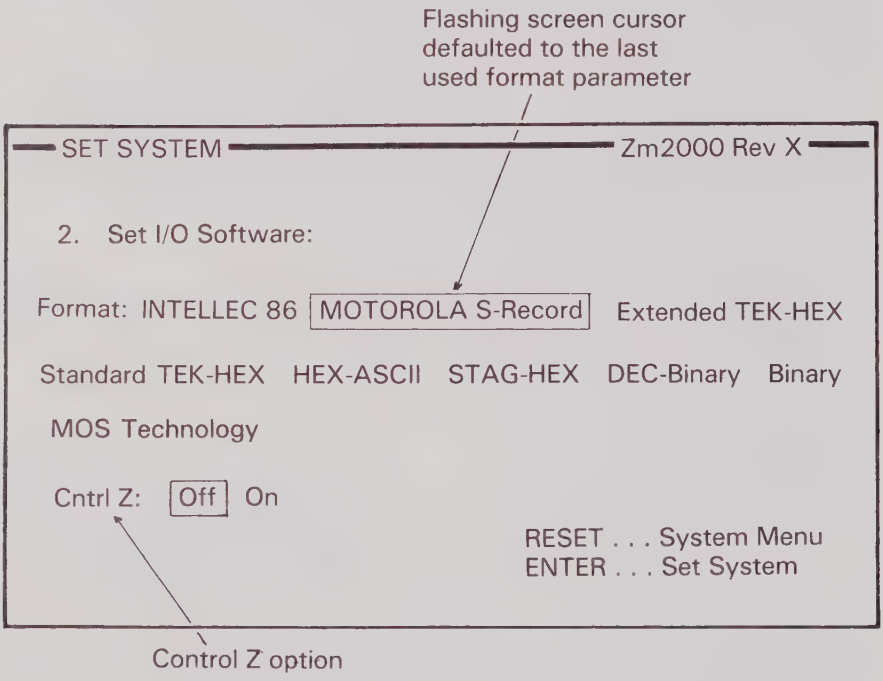
SET I/O SOFTWARE (Format Parameters)

Note: Format parameter selection for the PPZ does not apply to the IEEE interface. The IEEE may only be used under remote control, therefore the format selection must be made through the remote controller.

Selecting the format parameter.

From the SYSTEM MENU:

PRESS KEY A followed by KEY 2 to display:



Nine format parameters are available. Selection can be made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

The Control Z function is a development aid used mostly on Intel development systems. It is required to terminate the transmission of INPUT to an outside system.

The Control Z option can be selected by use of the down cursor and specified as either On or Off by use of the horizontal cursor keys.

To confirm the selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

6.2 I/O MODE

Only two of the available interface ports can be used for INPUT and OUTPUT when the PPZ is in the LOCAL MODE of operation: these are Serial I (The RS232C female connector) and Serial II (the RS232C male connector).

The IEEE-488 interface is a talker/listener port and may only be used under REMOTE CONTROL.

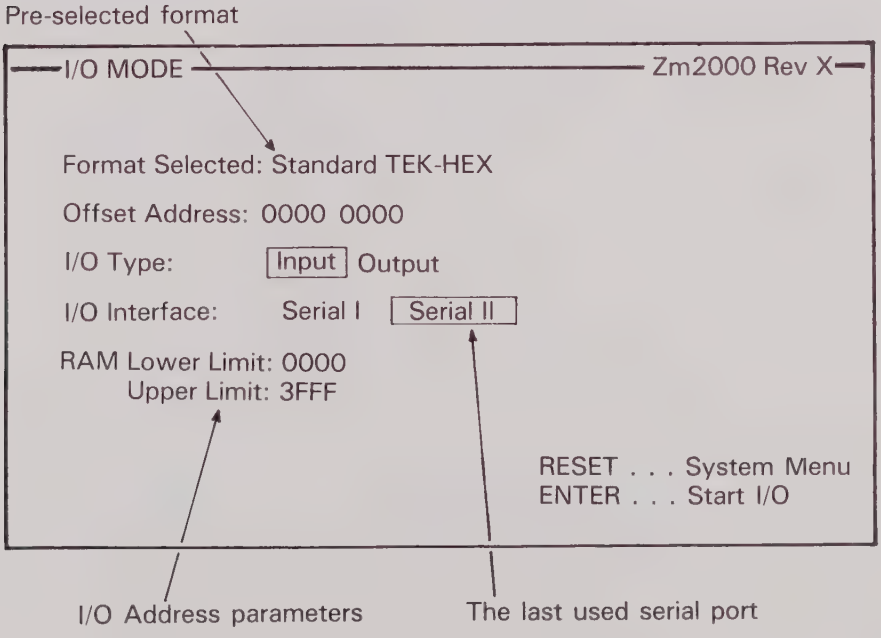
The RS232C ports can be individually selected to operate in either INPUT or OUTPUT modes although only the Serial I can be used for REMOTE CONTROL.

Selecting the I/O mode.

To enter the I/O mode from the SYSTEM MENU:

PRESS KEY I/O

This will display the associated parameters in their default states. For example:


Pre-selected format
I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0000
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 0000
Upper Limit: 3FFF
RESET . . . System Menu
ENTER . . . Start I/O
I/O Address parameters
The last used serial port

The display shows:

A. a flashing screen cursor indicating INPUT the I/O default state.

B. a static screen cursor indicating the last serial port used in conjunction with either INPUT or OUTPUT (in this example Serial II).

C. three pre-selectable address parameters for both INPUT and OUTPUT. These are:

INPUT OFFSET ADDRESS	1	OUTPUT OFFSET ADDRESS
INPUT RAM START ADDRESS	2	OUTPUT RAM START ADDRESS
INPUT RAM STOP ADDRESS	3	OUTPUT RAM STOP ADDRESS

The INPUT OFFSET ADDRESS is defaulted to zero.

The INPUT RAM START ADDRESS is defaulted to zero.

The INPUT RAM STOP ADDRESS is defaulted to the size of the RAM.

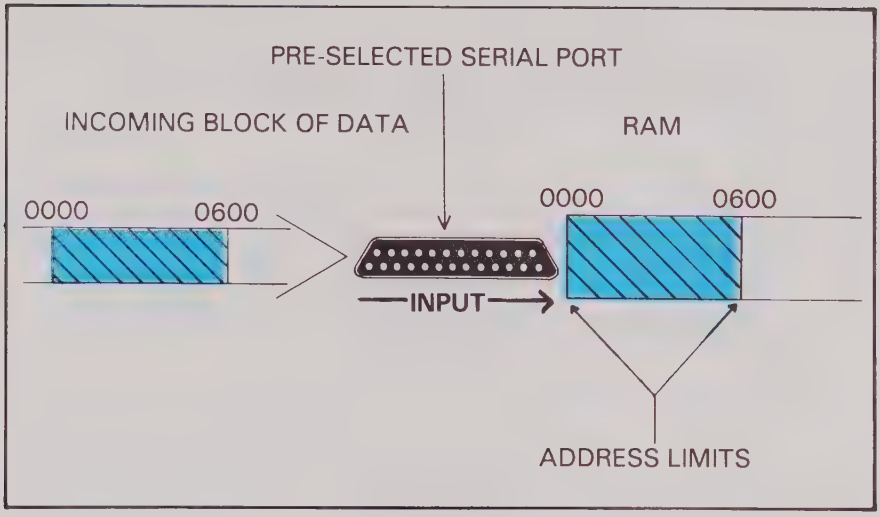
The OUTPUT OFFSET ADDRESS is defaulted to zero.

The OUTPUT RAM START ADDRESS is defaulted to zero.

The OUTPUT RAM STOP ADDRESS is defaulted to the size of the pre-selected device.

6.2.1 INPUT

Operation of INPUT

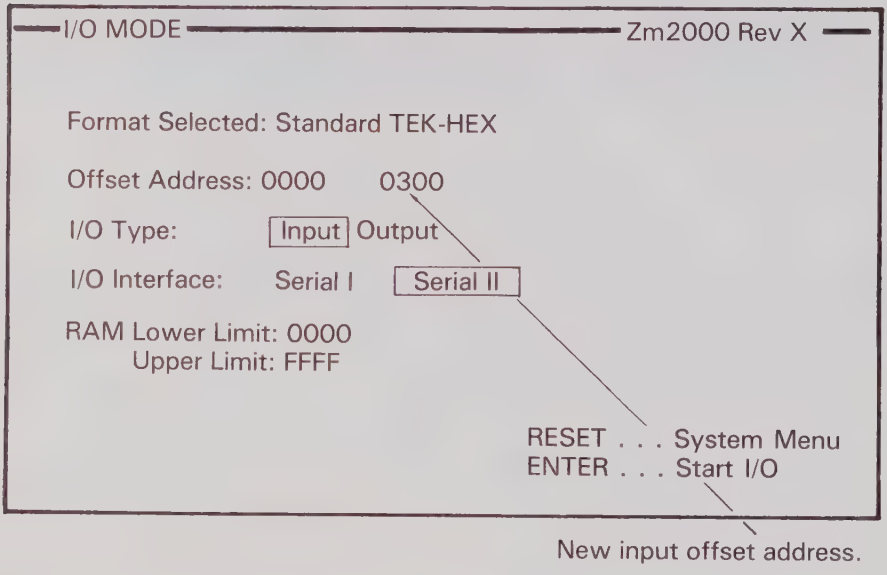


Input Offset Address

The offset address will be shown flashing at the top of the screen in its default state 0000 0000.

A new selection can be made by use of the hex-keyboard whereby up to eight figures can be directly entered onto the display.

For example:

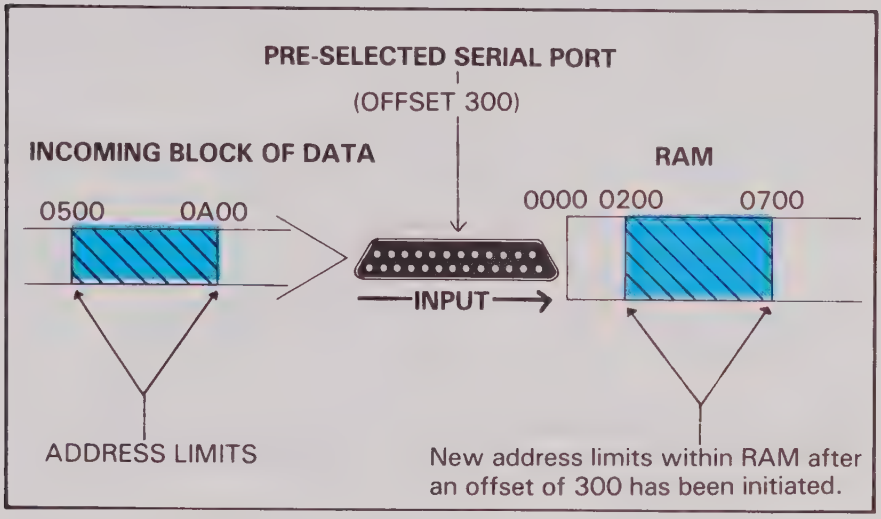

I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0300
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 0000
Upper Limit: FFFF
RESET . . . System Menu
ENTER . . . Start I/O
New input offset address.

Eight figures are available, in order to cater for the extended formats which contain a larger address field than the standard version. Therefore, to position data within the PPZ's RAM an eight figure offset may need to be specified.

Structure of Input Offset

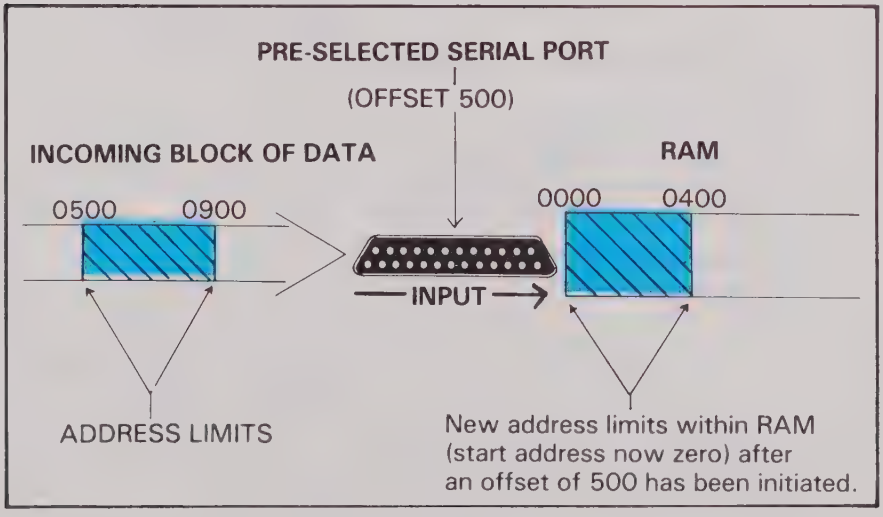
An incoming block of data originating from peripheral equipment can be re-located at a lower address within the RAM.

For example: OFFSET 0300



A more widely used example occurs when an offset is specified so that the start address of the incoming block of data becomes zero in the RAM.

For example:



Selecting Input and the Required Serial Port

If the offset address has been selected or is not required, to select the INPUT mode:

PRESS KEY ↓

The screen cursor will show the INPUT mode already entered. (INPUT is the default state of the I/O mode).

To select a specific serial port:

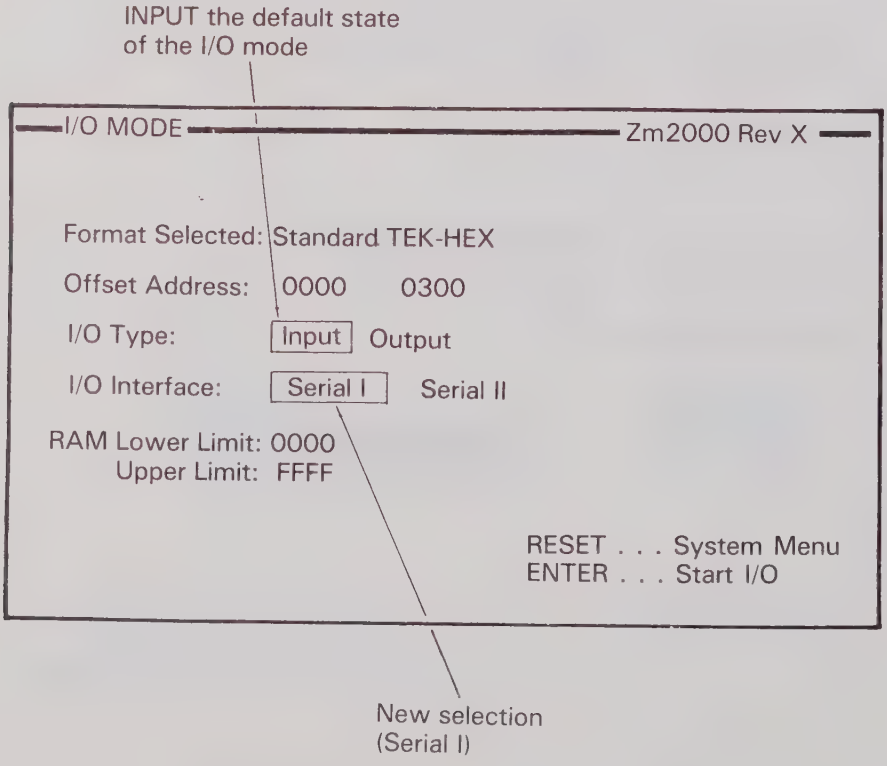
PRESS KEY ↓

This will show the screen cursor flashing on the last used selection (for instance Serial II).

To select Serial I:

PRESS KEY ←

The screen cursor will have now moved onto the Serial I selection. Interchange between the serial interfaces is made by use of both horizontal cursor keys.


INPUT the default state of the I/O mode
I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0300
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 0000
Upper Limit: FFFF
RESET . . . System Menu
ENTER . . . Start I/O
New selection (Serial I)

INPUT: RAM Start Address and RAM Stop Address

In addition to an offset a RAM start and stop address can be specified, whereby only a section of the in-coming data from peripheral equipment will be input into specified address limits within the RAM.

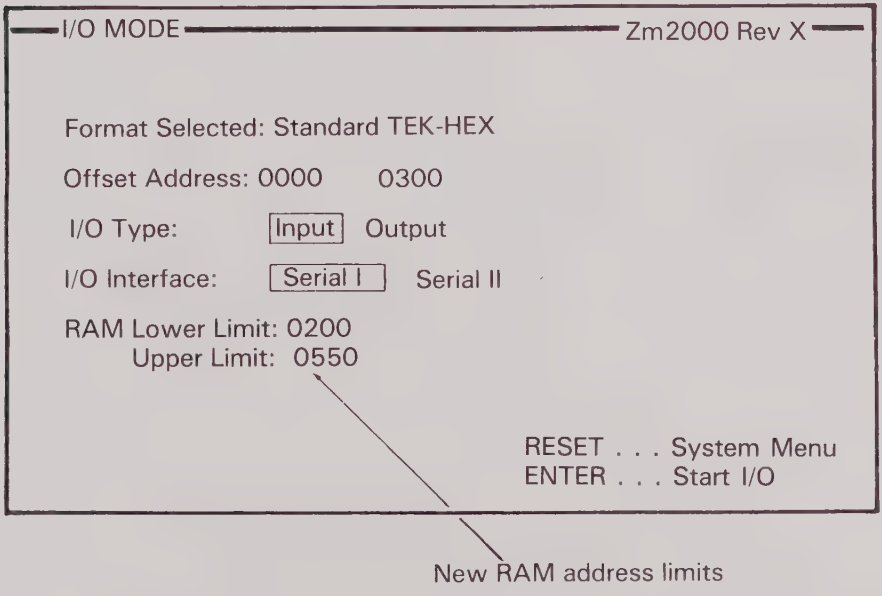
After the serial port has been specified, to select the RAM start and stop addresses:

PRESS KEY ↓

The lower limit will now be flashing. It will be in its default state of zero, and a new value can be entered from the hex-keyboard. To select the upper limit:

PRESS KEY ↓

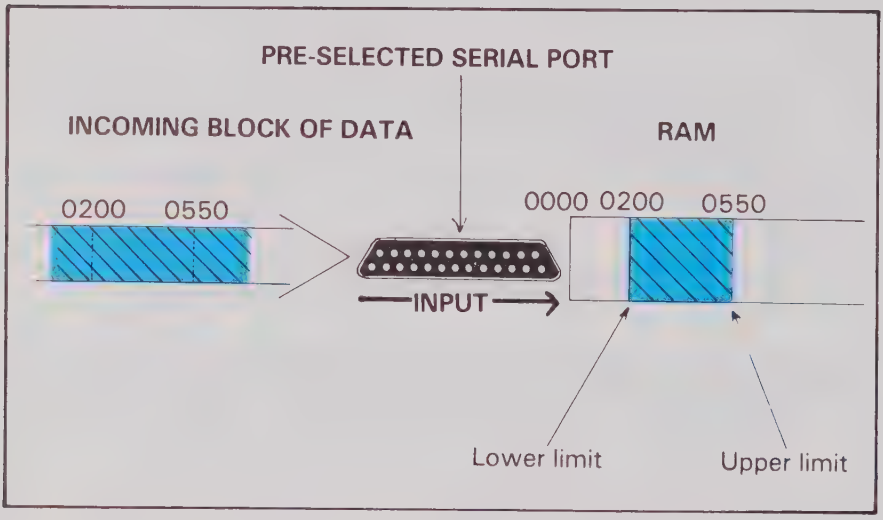
This will show the upper limit flashing. It will be defaulted to the size of the RAM. A new value can be entered from the hex-keyboard.


I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0300
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 0200
Upper Limit: 0550
RESET . . . System Menu
ENTER . . . Start I/O
New RAM address limits

Structure of Input Start and Stop Addresses

A section of the incoming data from peripheral equipment will be put into specified address limits within the RAM.

For example: Lower limit 0200
Upper limit 0550



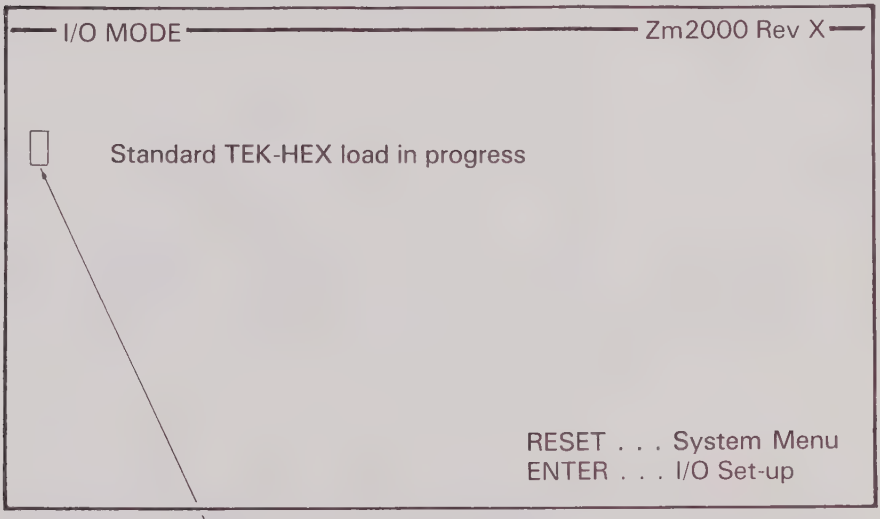
Implementing the Input

When the necessary parameters have been set, to implement INPUT:

PRESS ENTER

This will put the PPZ into a state of readiness to receive incoming data.

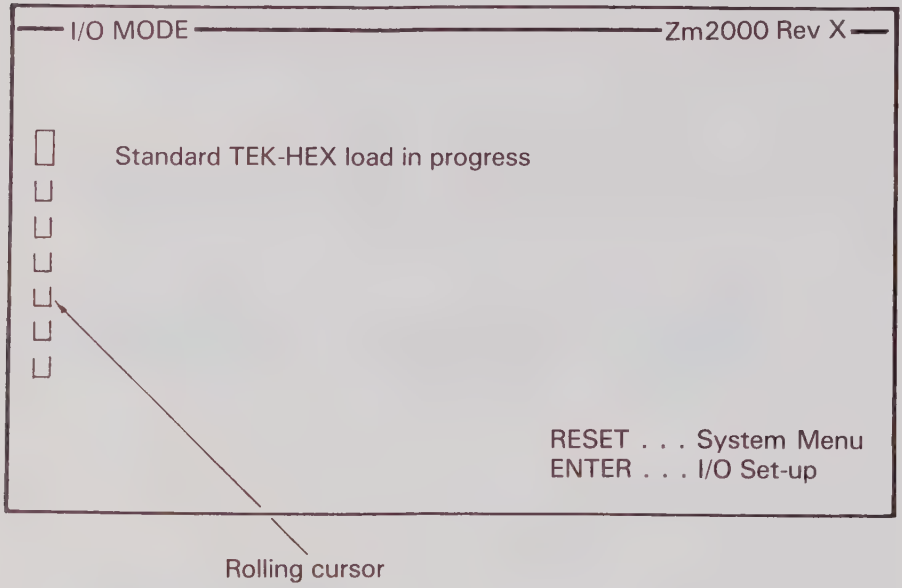
The display will show:



Static cursor, indicating that the PPZ is ready to receive incoming data.

During INPUT a cursor will roll in response to each ''carriage return, line feed'' (CR,LF) to indicate that the LOAD is in progress.

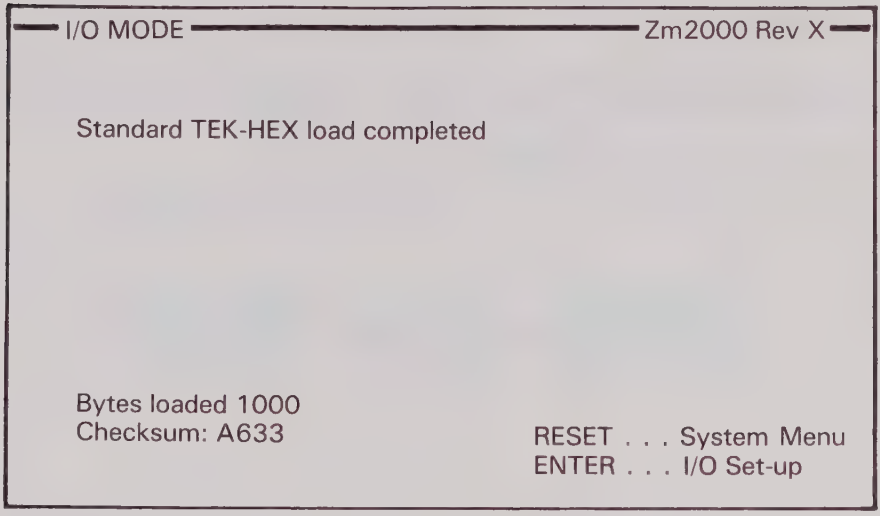
For example:



NB: Transmitted data that falls outside the boundaries of any pre-selected RAM limits will still be indicated during the INPUT operation by the rolling cursor but will not affect the RAM.

When the INPUT is complete the display will show a checksum and a byte count of all data loaded into RAM.

For example:



To return to the I/O mode:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Error Reporting on Input

Two possible error messages can be displayed by the PPZ at the end of a data INPUT instead of the checksum.

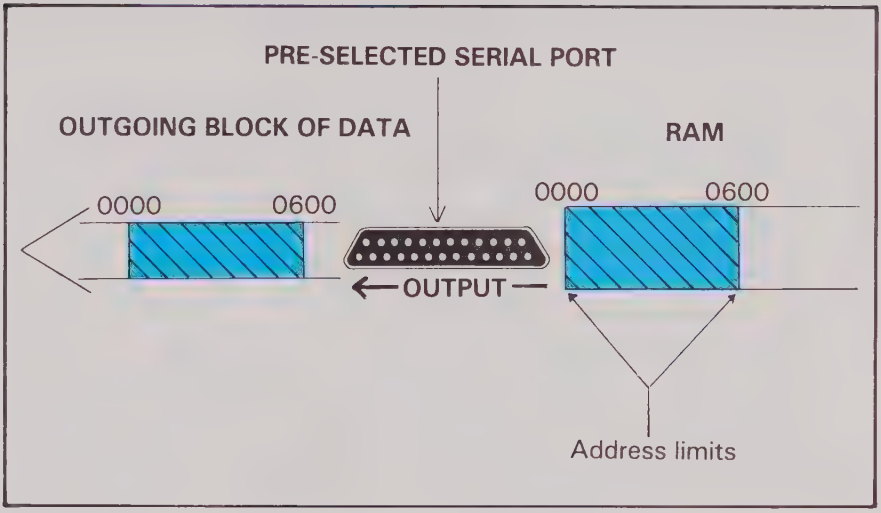
These are:

1. **Non-hex character received**
A non-hex character was received where a hex character was expected.

2. **Checksum field error detected.**
A checksum error in a record (a line) was detected.

6.2.2 OUTPUT

Operation of OUTPUT

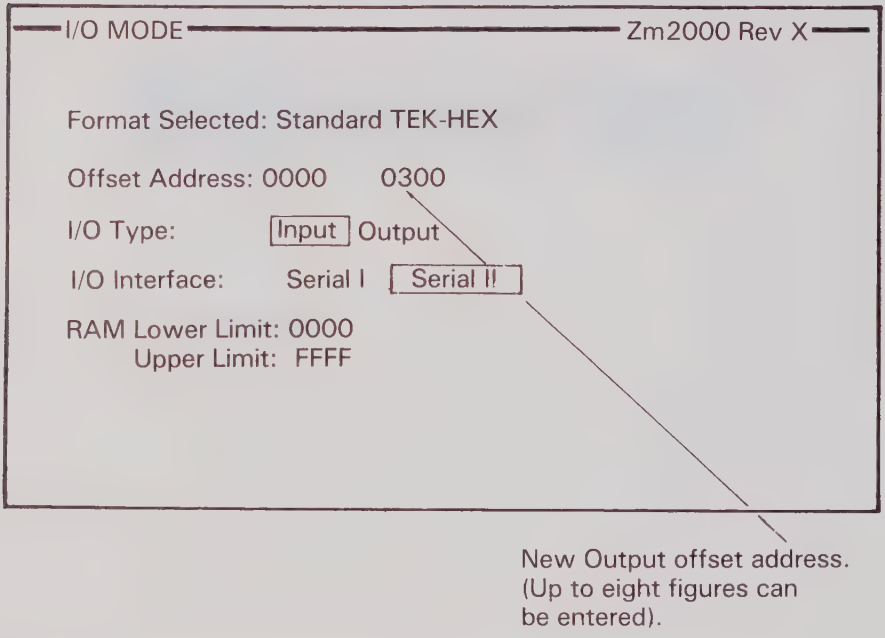


Output Offset Address

The offset address will be shown flashing at the top of the screen in its default state 0000 0000.

A new selection can be made by use of the hex-keyboard whereby up to eight figures can be directly entered onto the display.

For example:

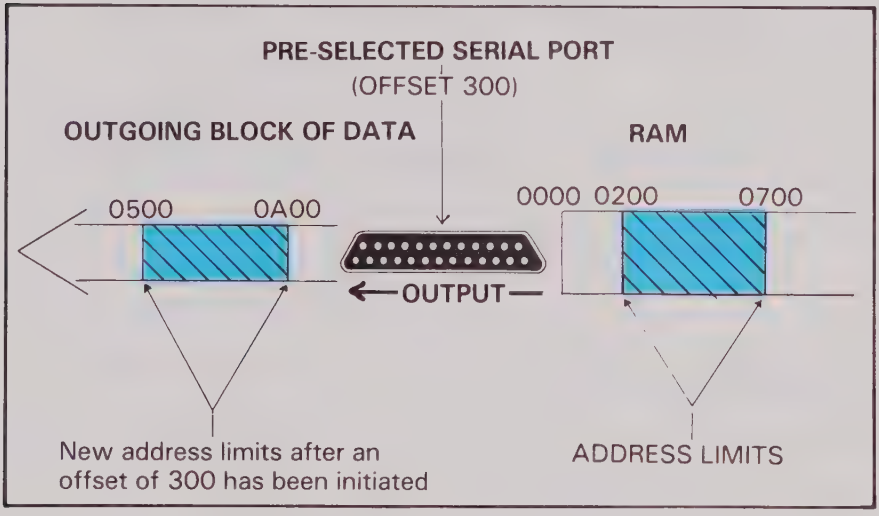

I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0300
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 0000
Upper Limit: FFFF
New Output offset address. (Up to eight figures can be entered).

Eight figures are available in order to cater for the extended formats which contain a larger address field than the standard versions. Therefore, to position data within the PPZ's RAM an eight figure offset may need to be specified.

Structure of Output Offset

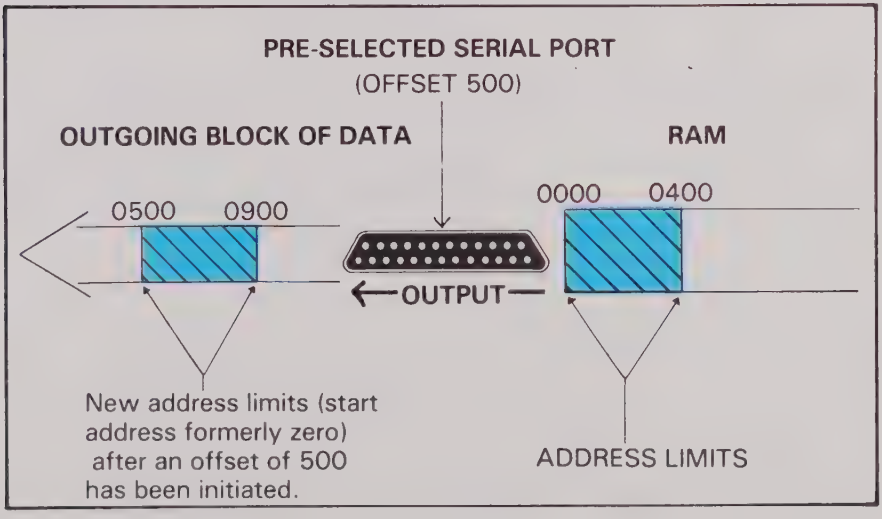
A block of data originating from the RAM can be re-located at a higher address.

For example: OFFSET 0300



A more widely used example is when re-location to a higher address occurs from a start address of 0000 in the RAM.

For example:



Selecting Output and the Required Serial Port

If the offset address has been selected or is not required, to select the OUTPUT mode:

PRESS KEY ↓

The screen cursor will show the INPUT mode which is the default state. To select OUTPUT:

PRESS KEY →

The screen cursor will now indicate that the PPZ is configured for this mode of operation.

To select a specific serial port:

PRESS KEY ↓

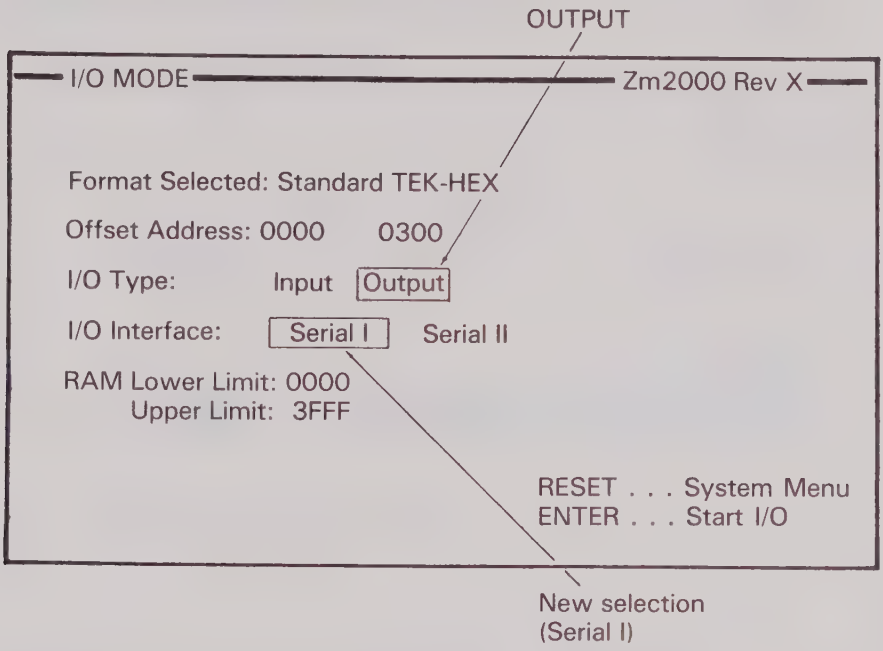
This will show the screen cursor flashing on the last used selection (for instance Serial II).

To select Serial I:

PRESS KEY ←

The screen cursor will have now moved onto the Serial I selection. Interchange between the serial interfaces is made by use of both horizontal cursor keys.

After having selected OUTPUT, the screen display will show the following:



In some circumstances, the static screen cursor indicating the last used serial port will move onto the alternative serial option simultaneously with selection of either INPUT or OUTPUT.

This feature simplifies the operation of the PPZ when it is linked (in LOCAL MODE) to two items of peripheral equipment via both serial interfaces.

For example: a ''computer'' (with the PPZ configured for INPUT through the serial I port) and a ''tape reader'' (with the PPZ configured for OUTPUT through the serial II port).

The configuration can be entered or reset by use of the cursor keys.

Output: RAM Start Address and RAM Stop Address

In addition to an offset, a RAM start and stop address can be specified on OUTPUT.

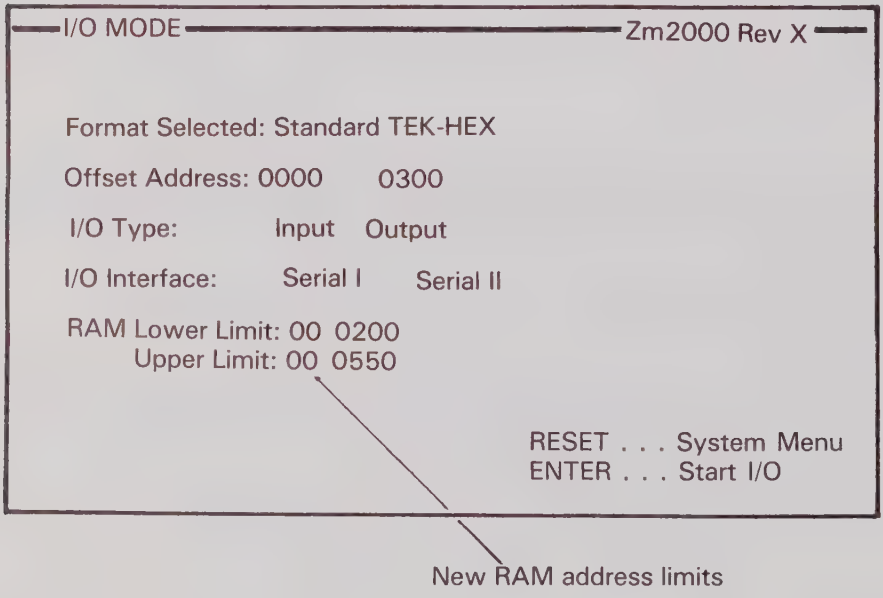
After the serial port has been specified, to select the RAM start and stop addresses:

PRESS KEY ←

The lower limit will now be flashing. It will be in its default state of zero, and a new value can be entered from the hex-keyboard. To select the upper limit:

PRESS KEY ←

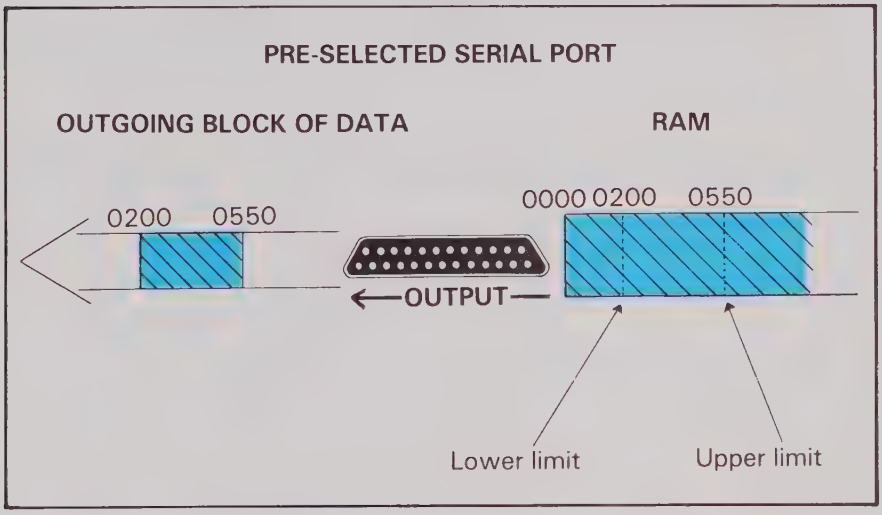
This will show the upper limit flashing. It will be defaulted to the size of the pre-selected device. A new value can be entered from the hex-keyboard.


I/O MODE
Zm2000 Rev X
Format Selected: Standard TEK-HEX
Offset Address: 0000 0300
I/O Type: Input Output
I/O Interface: Serial I Serial II
RAM Lower Limit: 00 0200
Upper Limit: 00 0550
RESET . . . System Menu
ENTER . . . Start I/O
New RAM address limits

Structure of Output Start and Stop Addresses

The block of data to be output within the RAM can be specified by using start and stop addresses.

For example: Lower Limit 0200
Upper Limit 0550



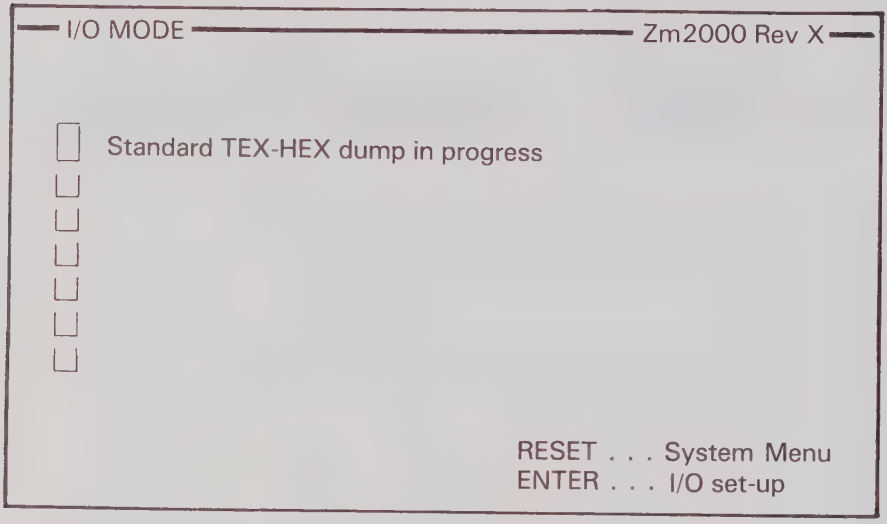
Implementing the OUTPUT

When the necessary parameters have been set, to implement the OUTPUT of data to peripheral equipment:

PRESS ENTER

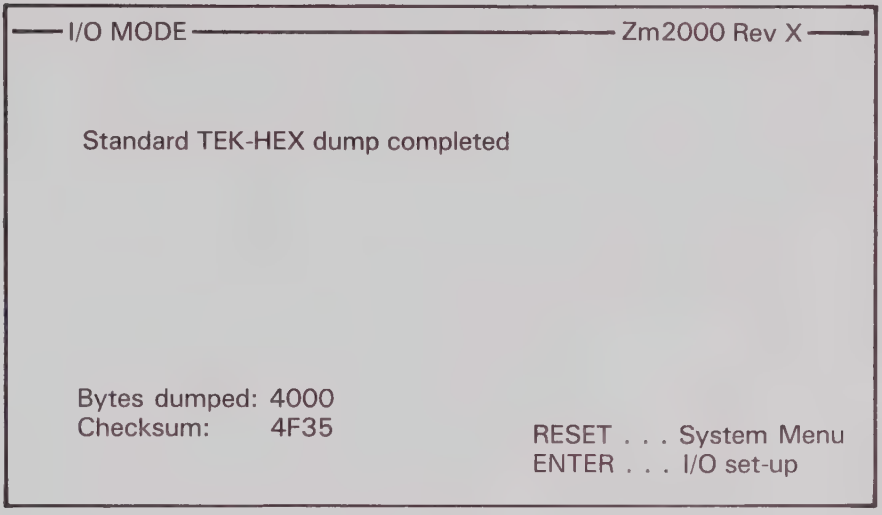
During OUTPUT a screen cursor will roll in response to each ''carriage return, line feed'' (CF, LF), ie. after each data line is sent from the RAM.

For example:



On completion of OUTPUT the display will show a checksum and a byte count of all data dumped.

For example:



To return to the I/O mode:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

SECTION 7

stag

Sophisticated systems for the discerning engineer.

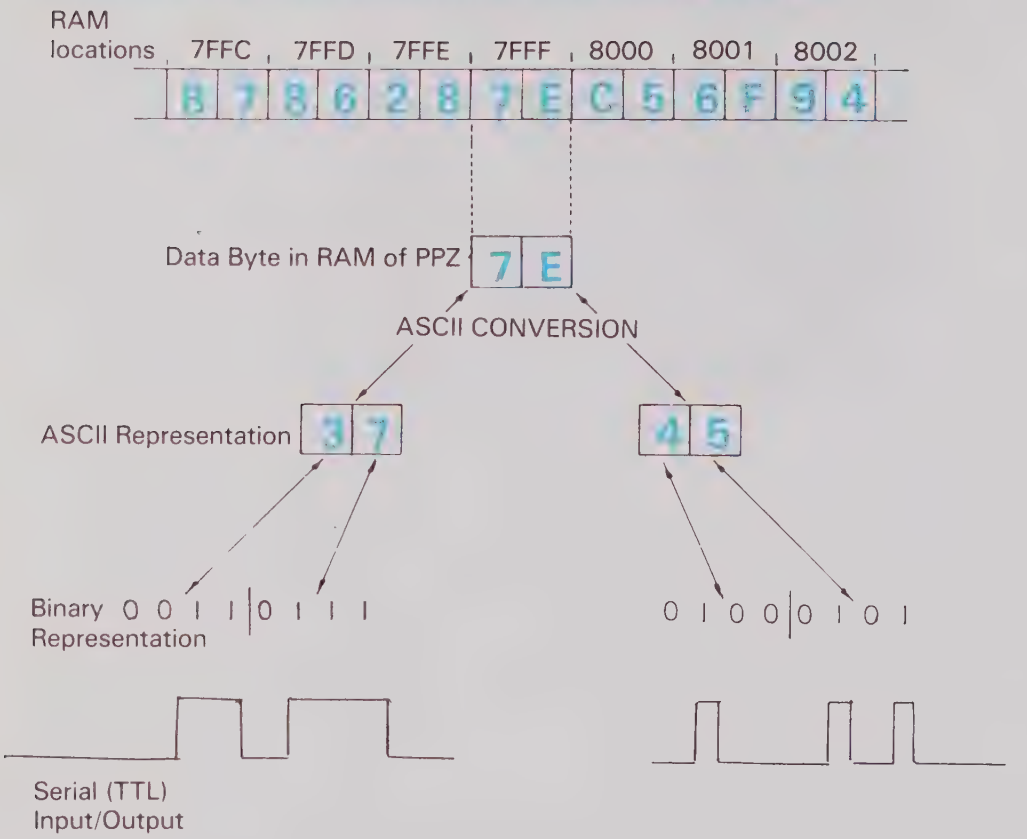
7.—FORMAT DESCRIPTIONS

7.1 INTERFACE FORMATS

The following formats are available on the PPZ

FORMAT	SECTION
STANDARD TEK-HEX	7.2
EXTENDED TEK-HEX	7.3
INTELLEC 86	7.4
MOTOROLA S-RECORD	7.5
HEX-ASCII	7.6
STAG-HEX*	7.7
DEC-BINARY/BINARY	7.8
MOS-TECHNOLOGY	7.9

STRUCTURE AND CONVERSION OF DATA BETWEEN THE SERIAL SIGNAL VIA THE RS232C INTERFACE PORTS TO THE PPZ RAM



7.2 STANDARD TEK-HEX

The TEK HEX format consists of:

(i) a start code, i.e. /

(ii) the address of the first byte of data in an individual record, e.g.: 0000

(iii) the sum of the number of data bytes in an individual record, e.g.: 1A

(iv) checksum 1 which is a nibble addition of the address (4 characters) and the byte count (2 characters), e.g.: 0B

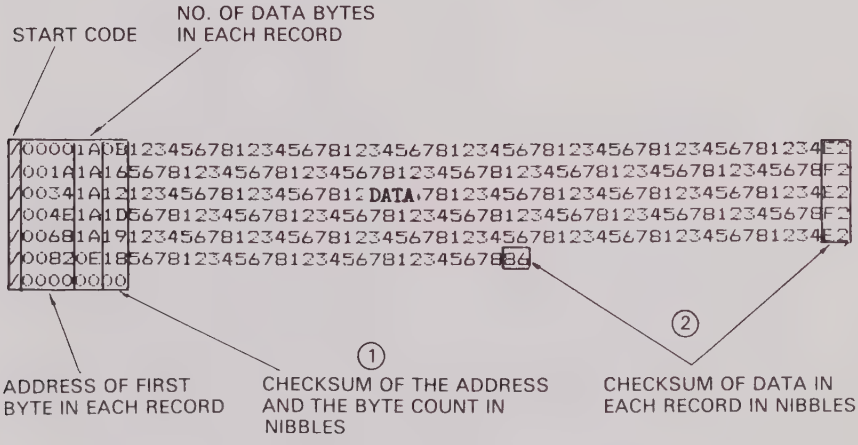
(v) data in bytes, e.g.: 12 34 56 78

(vi) checksum 2 which is a nibble addition of all data.

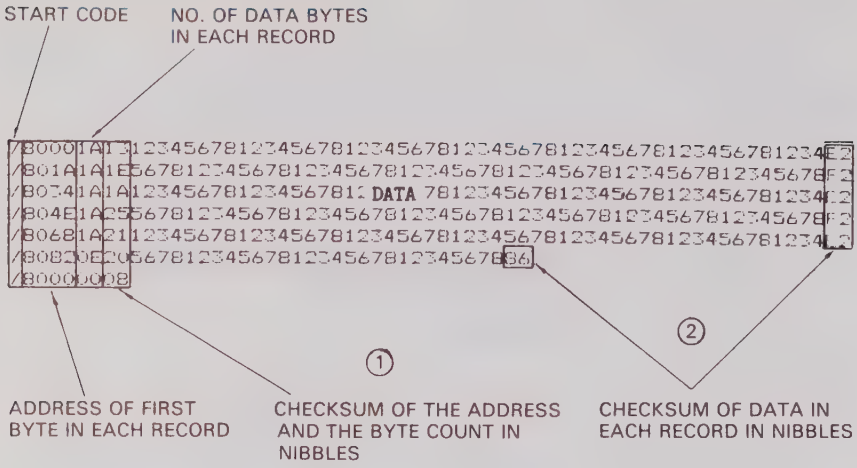
(vii) an end record which automatically stops the operation when 00 is specified in the byte count (iii).

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000



And with an Offset of (8000)


START CODE
NO. OF DATA BYTES IN EACH RECORD
DATA
ADDRESS OF FIRST BYTE IN EACH RECORD
CHECKSUM OF THE ADDRESS AND THE BYTE COUNT IN NIBBLES
CHECKSUM OF DATA IN EACH RECORD IN NIBBLES
1
2

Calculation of Tek Hex Checksums

Unlike the other PPZ formats, the TEK HEX has two checksums which are both the result of nibble additions, as opposed to byte additions.

Checksum 1 is a nibble addition of the 'address' and the 'byte count' which make 6 characters in total.

Checksum 2 is a nibble addition of data alone.

```
/00001A0B12345678123456781234567812345678123456781234567812345678123456781234E2
CHECKSUM 1 CHECKSUM 2 678123456781234567812345678123456781234567812345678F2
/10034030A12345615
/00000000
```

Example: The third ''Data Record'' of the above format.

CHECKSUM 1

(i)	this is:	/10034030A
(ii)	the start code and the checksum are removed:	/0A
(iii)	six nibbles remain:	003403
(iv)	they are added together:	0+0+3+4+0+3 = A
(v)	A is the checksum which is displayed in byte form as above:	/1003403 (0A)

CHECKSUM 2

(i)	this is	123456 15
(ii)	the checksum is removed:	15
(iii)	six nibbles remain:	123456
(iv)	these are added together:	1+2+3+4+5+6 = 15
(v)	15 is the checksum as above:	123456 (15)

When addition of nibble information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.3 EXTENDED TEK HEX

The EXTENDED TEK HEX consists of:

(i) a start code: % (percentage)

(ii) a count of the nibbles in an individual record, e.g.: 3B

(iii) the record types, i.e.: 6—Data Record;
8—End Record

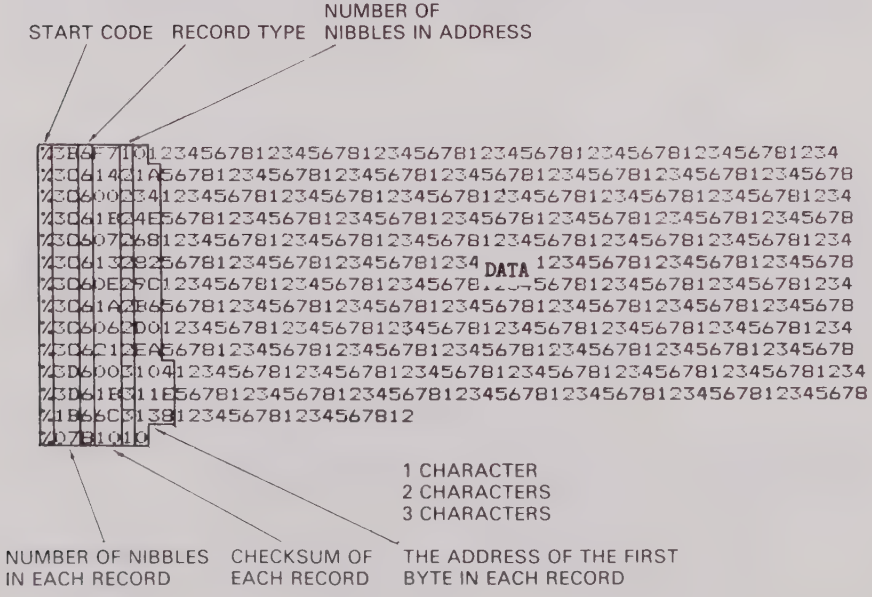
(iv) a checksum of the whole of an individual record excluding the %
e.g.: F7

(v) *the number of nibbles comprising—''the address of the first byte in each record'', e.g.: 1, 2, 3 etc.

(vi) the address of the first byte of data in an individual record, e.g.: 0, 1A, 104

For example:

START ADDRESS:	0000
STOP ADDRESS:	0140
OFFSET:	0000 0000



*Sections (e) and (f) are integrated:
As the operation progresses the address field lengthens. More characters are added to show this expansion. The nibble count of section (e) reflects this, e.g.:

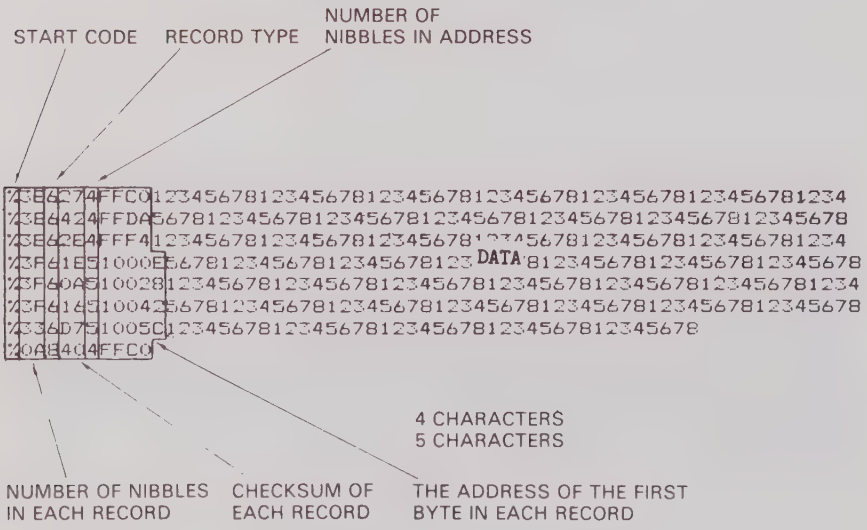
2/1A	6/100000	A/1B4625DC95
2 Characters	6 Characters	A Characters (10 in Decimal)

The nibble count takes values from '0' to rise to 'F' making a fifteen character address field possible.

EXTENDED TEK HEX WITH AN OFFSET, DISPLAYING TRANSLATION FROM 4 CHARACTER ADDRESS FIELD TO 5 CHARACTER ADDRESS FIELD.

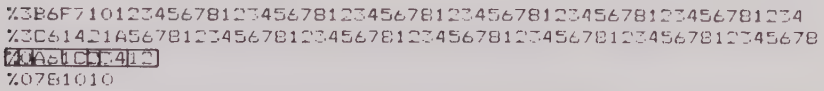
For example:

START ADDRESS:	0000
STOP ADDRESS:	00AF
OFFSET:	0000 FFC0



Calculation of the Extended Tek Hex Checksum

Unlike the standard version the EXTENDED TEK HEX has only one checksum.



Example: The third line of the above format.

(i)	this is:	% 0A61C23412
(ii)	the start code and the checksum are removed:	% IC
(iii)	eight nibbles remain:	0A623412
(iv)	these are added together:	0 + A + 6 + 2 + 3 + 4 + 1 + 2 = 1C
(v)	1C is the checksum as above:	% 0A6 1C 23412

When addition of nibble information occurs in longer records, the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.4 INTELLEC 86

The INTELLEC 86 format consists of:

(i) a start code, i.e.: (colon)

(ii) the sum of the number of bytes in a particular record, e.g.: 1A

(iii) the address of the first byte of data in an individual record, e.g.: 0000

(iv) the record types, i.e.: 00—Data Record
01—End Record
02—Segment Base Address record (SBA)*

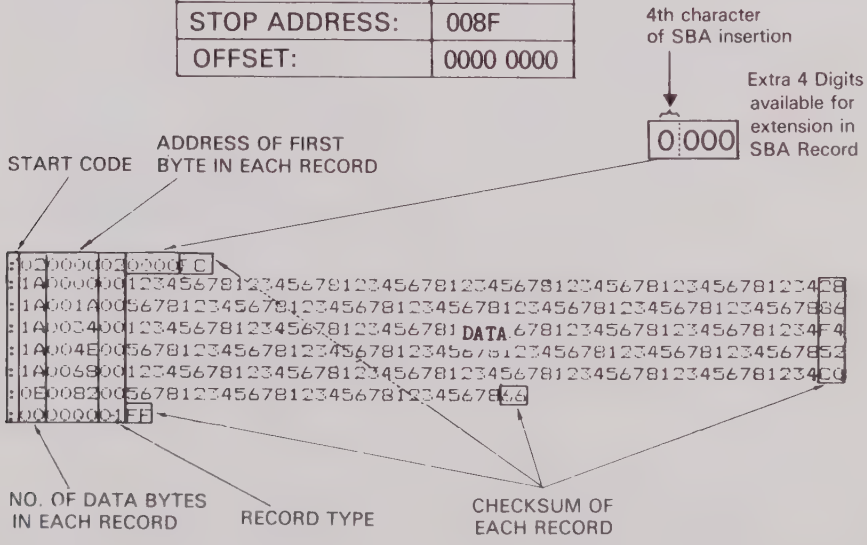
*The SBA is the record that displays the INTELLEC 86 extension. This is achieved by the provision of an extra digit which corresponds to the 4th character of the SBA insertion. This 4th character is effectively the extension which lengthens the standard (FFFF) limitation, into the EXTENDED INTELLEC (FFFFF).

(v) data (in bytes), e.g.: 12 34 56 78

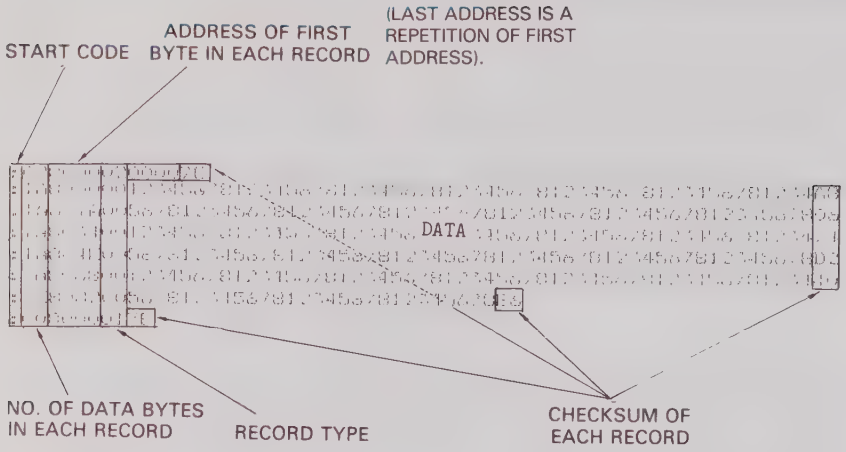
(vi) a checksum of an individual record, e.g.: 28

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000 0000


4th character of SBA insertion
Extra 4 Digits available for extension in SBA Record
0000
START CODE
ADDRESS OF FIRST BYTE IN EACH RECORD
DATA
NO. OF DATA BYTES IN EACH RECORD
RECORD TYPE
CHECKSUM OF EACH RECORD

And with an Offset of: (0000 8000)


START CODE
ADDRESS OF FIRST BYTE IN EACH RECORD
(LAST ADDRESS IS A REPETITION OF FIRST ADDRESS).
DATA
NO. OF DATA BYTES IN EACH RECORD
RECORD TYPE
CHECKSUM OF EACH RECORD

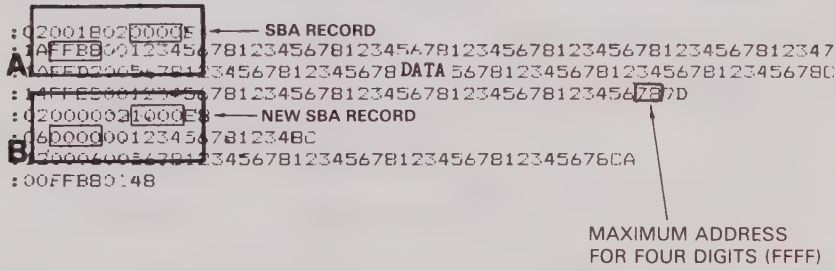
SBA Repetition

In some operations where an offset is in use the SBA can be displayed twice.

When the address field passes the maximum quantity for a four digit figure, i.e.: (FFFF) a second SBA record is specified.

For example:

START ADDRESS	FFAO
STOP ADDRESS	FFFF
OFFSET	0000 0018



The SBA is added to the address field in the following fashion:

EXTENSION DIGIT	B
1 0 0 0	SBA INSERTION
+ 0 0 0 0	ADDRESS FIELD
= 1 0 0 0 0	

EXTENSION DIGIT	A
0 0 0 0	SBA INSERTION
+ F F B 8	ADDRESS FIELD
= 0 F F B 8	

If required by the user the remaining 3 digits of the SBA insertion can be non zero.

Calculation of the Intellec 86 Checksum

```
:1A000000123456781234567812345678123456781234567812345678123428
:0100001A0056 8F
:00000001FF
```

Example: The second "Data Record" of the above format.

(i)	this is:	: 01 00 1A 00 56 8F	
(ii)	the start code and the checksum are removed:	: 8F	
(iii)	five Bytes remain:	01 00 1A 00 56	
(iv)	these are added together:	01 + 00 + 1A + 00 + 56 = 71	
(v)	the total '71' is converted into Binary:	7 0111	1 0001
(vi)	the Binary figure is reversed. This is known as a complement:	8 1000	E 1110
(vii)	a one is added to this complement. This addition forms a two's complement	8 1000	F 1111
(viii)	8F is the checksum as above:	: 01 00 1A 00 56 8F	

When addition of information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.5 MOTOROLA S-RECORD

The MOTOROLA S-RECORD is identical to its standard version when displayed, up to the point that the data's address goes beyond FFFF and thus requires a 5th digit, e.g.: 10000. To compensate for this addition an extra byte is added to the address giving 010000.

When this occurs the record type changes:

The data record changes from 1 to 2
and the end record changes from 9 to 8.

Similarly when the data address goes beyond FFFFFF a 7th digit is required and likewise a byte is added giving the address 8 characters: 01000000.

When this occurs:

The data record changes from 2 to 3
and the end record changes from 8 to 7.

The MOTOROLA S-RECORD consists of:

(i) a start code, i.e.: S

(ii) the record types, i.e.: 1—Data Record (Four Character address)
9—End Record (Four Character address)

2—Data Record (Six character address)
8—End Record (Six character address)

3—Data record (Eight character address)
7—End Record (Eight character address)

(iii) The sum of the number of bytes in an individual record, e.g.: 1D

(iv) the address of the first byte of data in an individual record, e.g.: 0000, 010000, 01000000

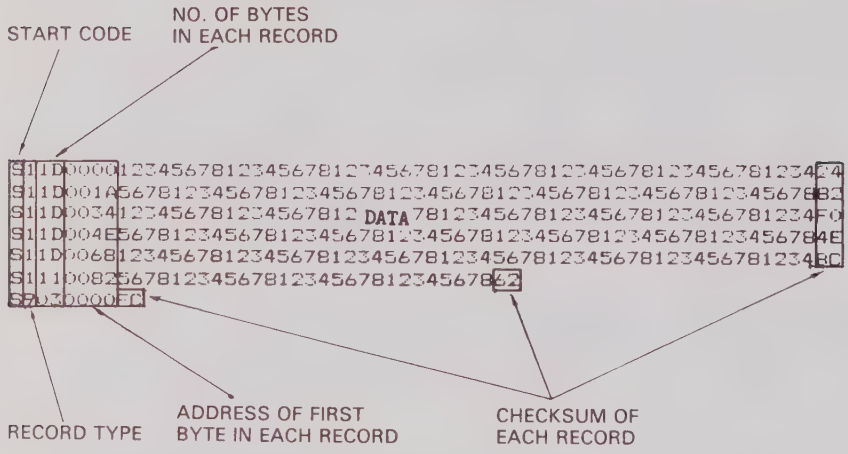
(v) data in bytes, e.g.: 12 34 56 78

(vi) checksum of an individual record: 24

1-Data Record (Four Character Address)
9-End Record (Four Character Address)
} 2 BYTES

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000 0000

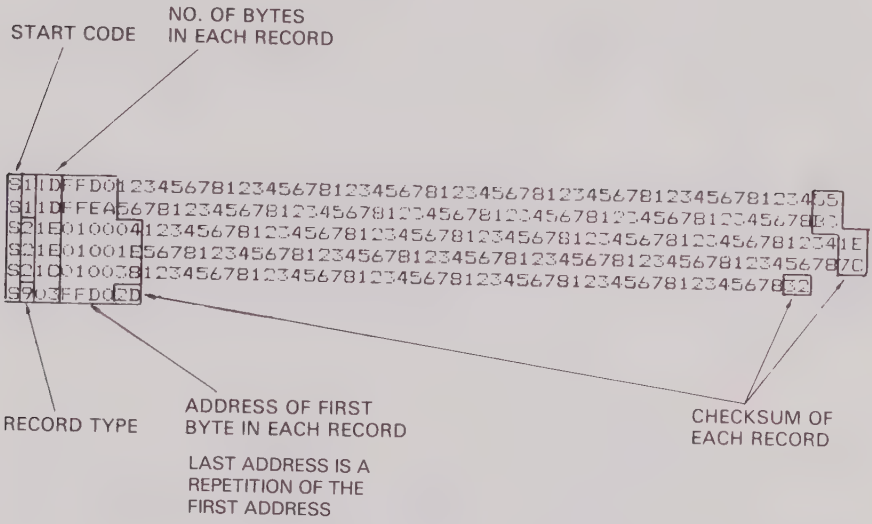


1—Data Record (Four Character Address)
9—End Record (Four Character Address)
} 2 BYTES

The MOTOROLA S-RECORD format stays identical in layout to that of the standard when the address field stays below FFFF.

Translation from 2 Byte Address (4 Characters) Through to 3 Byte Address (6 Characters).

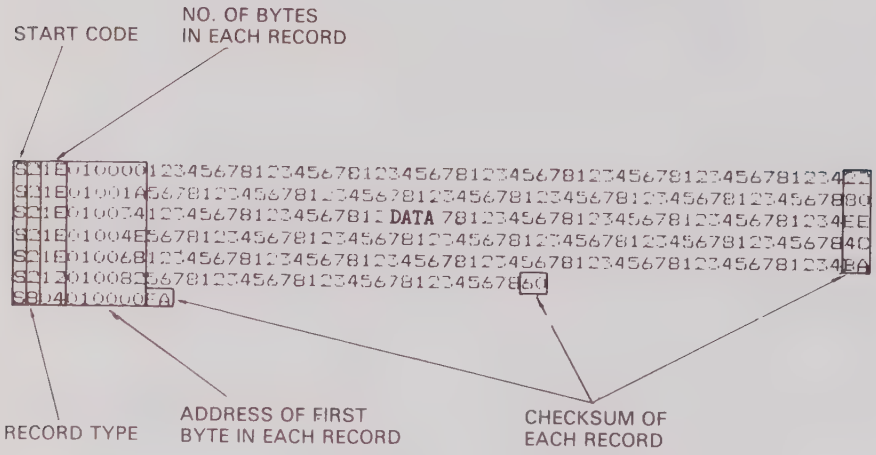
START ADDRESS:	FF80
STOP ADDRESS:	FFFF
OFFSET:	00000050


START CODE
NO. OF BYTES IN EACH RECORD
RECORD TYPE
ADDRESS OF FIRST BYTE IN EACH RECORD
LAST ADDRESS IS A REPETITION OF THE FIRST ADDRESS
CHECKSUM OF EACH RECORD

2-Data Record (Six Character Address)
8-End Record (Six Character Address) } 3 BYTES

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	00010000

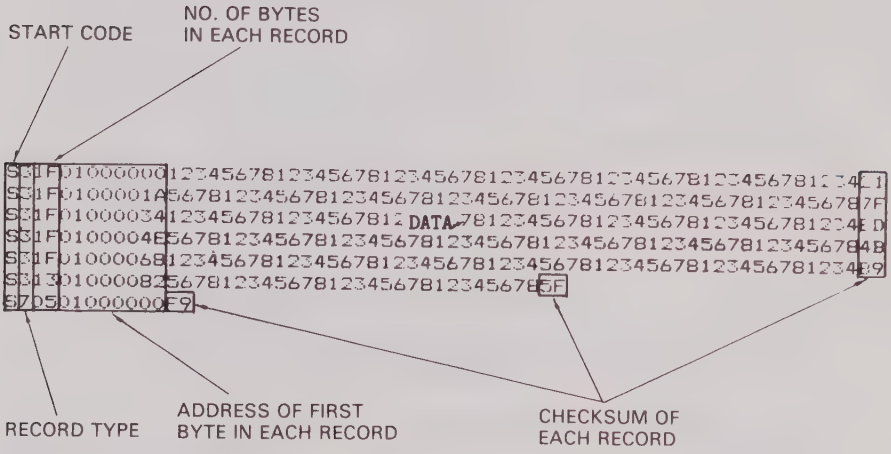


2-Data Record (Six Character Address)
8-End Record (Six Character Address) } 3 BYTES

3-Data Record (Eight Character Address)
7-End Record (Eight Character Address) } 4 BYTES

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	01000000



3-Data Record (Eight Character Address)
7-End Record (Eight Character Address) } 4 BYTES

Example: The second "Data Record" of the above format.

(i)	this is:	S1 04 00 1A 56 8B
(ii)	the start code, the record type and the checksum are removed:	S1 8B
(iii)	four bytes remain:	04 00 1A 56
(iv)	these are added together:	04 + 00 + 1A + 56 = 74
(v)	the total '74' is converted into Binary:	7 = 0111, 4 = 0100
(vi)	the Binary figure is reversed. This is known as a complement*:	8 = 1000, B = 1011
(vii)	8B corresponds to the checksum as above:	S1 04 00 1A 56 (8B)

When addition of information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

*When no additional figures are added to this calculation it is called a one's complement.

7.6 HEX ASCII

The HEX ASCII format consists of:

DATA ALONE

However, invisible instructions are neccessary for operation. These are:

(i) a hidden start character known as Control B.
(02: ASCII Code, STX ASCII character).

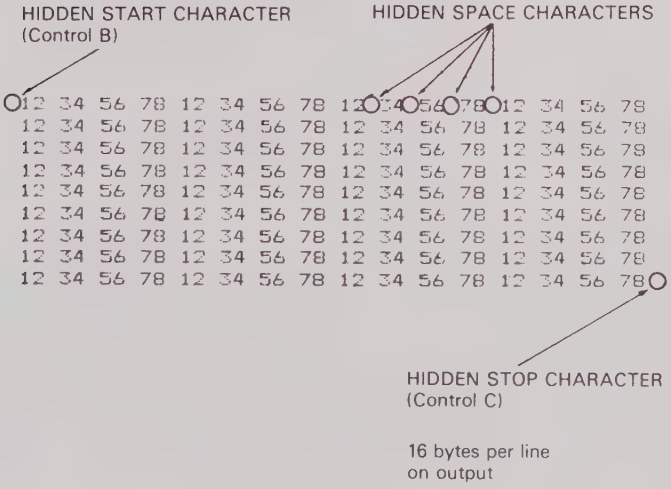
(ii) a hidden stop character known as Control C.
(03: ASCII Code, EXT ASCII character).

(iii) a hidden 'space' character between data bytes
(20: ASCII Code, SP ASCII character)

For example:

START ADDRESS	0000
STOP ADDRESS:	008F

OFFSET: NONE REQUIRED AS
HEX ASCII ALWAYS LOADS AT ZERO



16 bytes per line
on output

7.7 STAG HEX*

The STAG HEX* format consists of:

(i) a start code, i.e. * (an asterisk, 2A—ASCII Code)

(ii) the address of the first byte of data in an individual record, e.g. 0000

(iii) data in bytes, e.g. 12 34 56 78

(iv) a stop code, i.e. $ (a dollar sign, 24—ASCII Code)

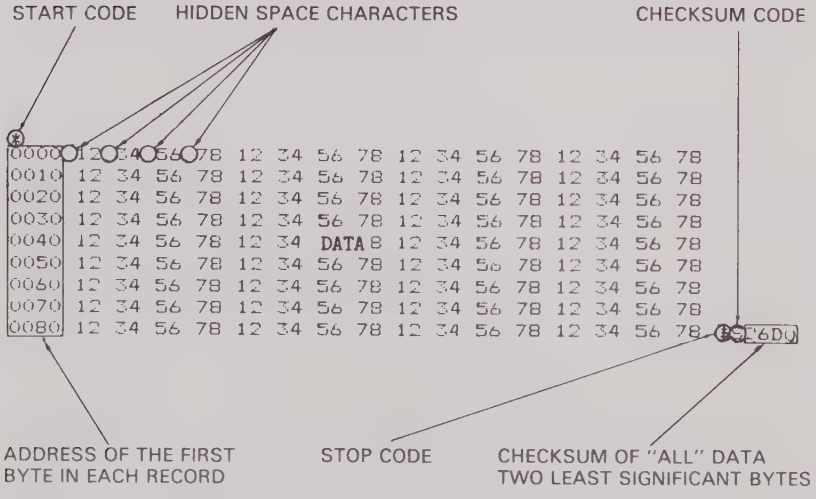
(v) a checksum of all data over the entire address range. (The checksum is the two least significant bytes.)

(vi) a checksum start code: S

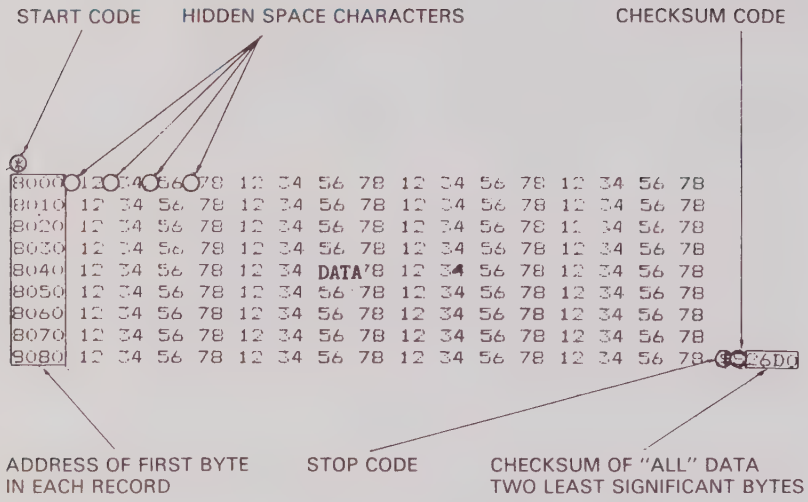
(vii) an invisible space character between data bytes (20-ASCII Code)

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000



And with an Offset of 8000



Calculation of the STAG HEX* Checksum

''Data alone'', in bytes over the entire address range (as opposed to individual records) is added together to give the checksum. The address is not included in this calculation.

```
*
0000 12 34 56 78 $S0114
```

Example: The segment of data above.

(i)	this is:	*0000 12 34 56 78 $S0114
(ii)	the start code, the address, the stop code, the checksum code and the checksum are removed:	*0000 $S0114
(iii)	four Bytes remain:	12 34 56 78
(iv)	these are added together:	12 + 34 + 56 + 78 = 114
(v)	114 is the checksum which is displayed in two byte form as above:	*0000 12 34 56 78 $S0114

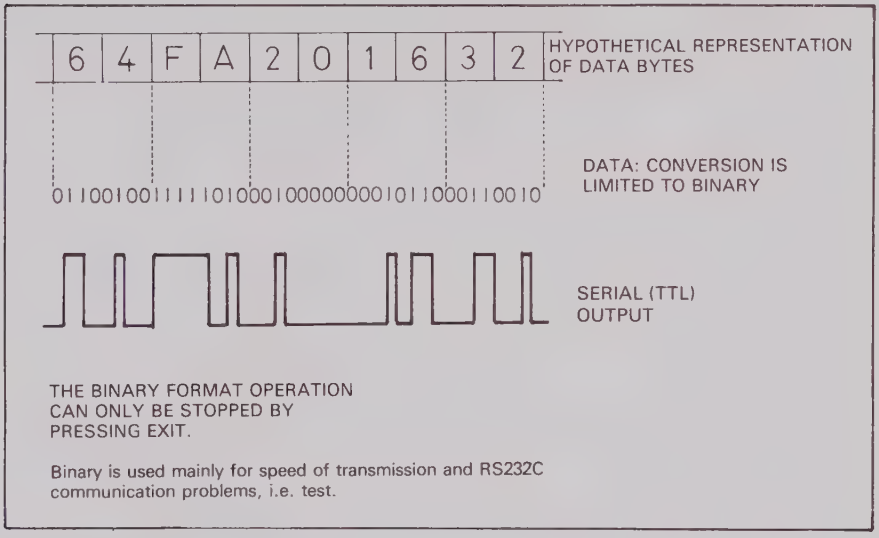
As the STAG HEX* checksum is an addition of all data the total will invariably constitute more than two bytes. When this occurs the least significant 'two' bytes are always selected to undergo the above calculation.

7.8 BINARY AND DEC BINARY

Binary and DEC Binary are the most fundamental of all formats. ASCII code conversion never occurs. Information is therefore limited to the interpretation of pulses via the RS232C interface port into either ONE'S or ZERO's. Hence 'Binary'. A visual display is not possible.

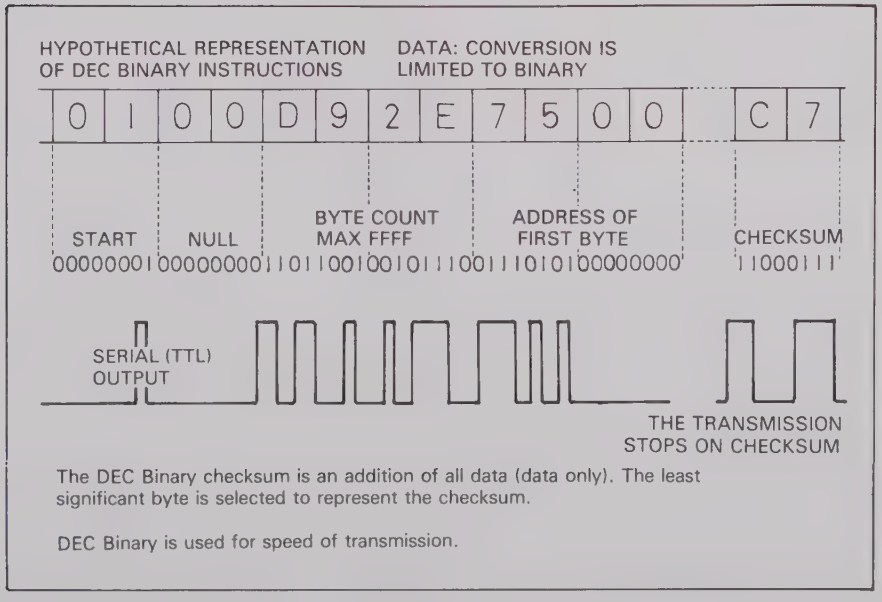
BINARY:

Binary is data only. It is devoid of a start code, address, stop code and checksum.


6 4 F A 2 0 1 6 3 2
HYPOTHETICAL REPRESENTATION OF DATA BYTES
DATA: CONVERSION IS LIMITED TO BINARY
SERIAL (TTL) OUTPUT
THE BINARY FORMAT OPERATION CAN ONLY BE STOPPED BY PRESSING EXIT.
Binary is used mainly for speed of transmission and RS232C communication problems, i.e. test.

DEC BINARY

DEC Binary is an improvement of Binary. It has a start code, a null prior to transmission, a byte count, a single address and a single checksum of all data. It also has the facility for an offset to be set.


HYPOTHETICAL REPRESENTATION OF DEC BINARY INSTRUCTIONS
DATA: CONVERSION IS LIMITED TO BINARY
0 1 0 0 D 9 2 E 7 5 0 0 C 7
START
NULL
BYTE COUNT MAX FFFF
ADDRESS OF FIRST BYTE
CHECKSUM
SERIAL (TTL) OUTPUT
THE TRANSMISSION STOPS ON CHECKSUM
The DEC Binary checksum is an addition of all data (data only). The least significant byte is selected to represent the checksum.
DEC Binary is used for speed of transmission.

7.9 MOS-TECHNOLOGY

The MOS-TECHNOLOGY format consists of:

i. a start code, ie: ; (semi-colon)

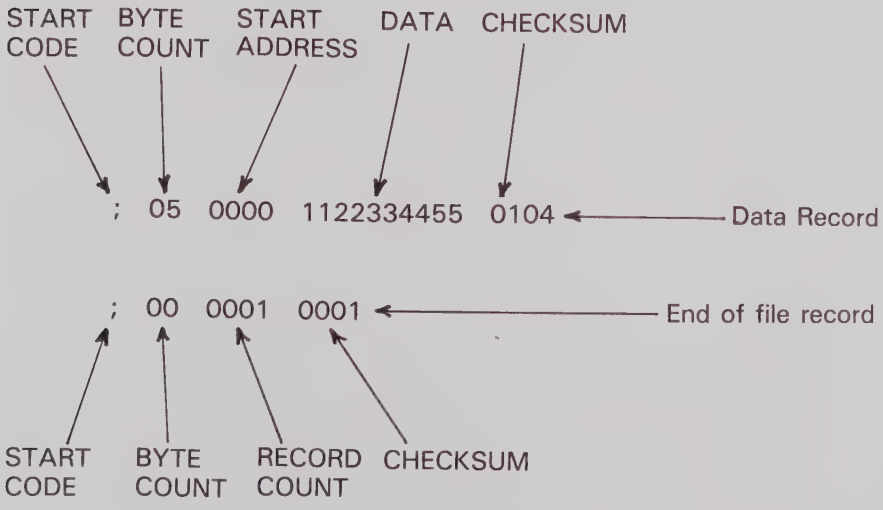
ii a byte count—that is the sum of the number of data bytes in an individual record, eg: 05

iii the address of the first byte of data in an individual record, eg: 0000

iv data in bytes, eg: 11 22 33 44 55. (The data bytes must consist of valid hexadecimal digits).

v. a checksum which is displayed as two hexadecimal bytes. It is the addition of the preceeding data bytes in the record including the address and byte count in hexadecimal form.

For example:



Calculation of MOS-Technology checksum

```
; 05 0000 11 22 33 44 55 0104
; 00 0001 0001
```

Example: the first line of the above format.

i. this is ; 05 0000 11 22 33 44 55 0104

ii. the start code and the checksum are removed: ; 0104

iii. this leaves:
the byte count: 05
the address of the first byte in the record: 0000
and five data bytes: 11 22 33 44 55

iv. these are added together:
05 + 000 + 11 + 22 + 33 + 44 + 55 = 0104

v. 0104 is the checksum as above: ;050001122334455 0104

SECTION 8

stag

Sophisticated systems for the discerning engineer.

8. — RS232C HARDWARE DESCRIPTIONS

8.1 RS232C INTERFACE PORT CONNECTIONS

The PPZ can be linked to peripheral equipment via one or both of its RS232C interface ports.

RS232(1) Serial I Female connector

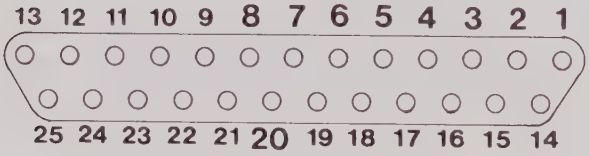
The RS232(1) port is a D type female connector configured as DTE (Data Terminal Equipment) which is able to accept a 25 way male connector (DCE).

It is generally used to connect the PPZ to a computer system or a modem.

In 'local' or 'remote' mode it can be used to load data to the PPZ RAM or dump data from the PPZ RAM to peripheral equipment.

Only 9 of the 25 available pin sockets play an active role in data transfer.

These are: numbers 1, 2, 3, 4, 5, 6, 7, 8 and 20.



Individual Pin Functions on the Serial I Connection

1.*	PG	PROTECTIVE GROUND
2.	TXD	TRANSMITTED DATA
3.	RXD	RECEIVED DATA
4.	RTS	REQUEST TO SEND
5.	CTS	CLEAR TO SEND
6.	DSR	DATA SET READY
7.	SG	SIGNAL GROUND
8.	DCD	DATA CARRIER DETECT
20.	DTR	DATA TERMINAL READY

*Pin number 1 is present in all connections, it represents the protective ground surrounding all the other cables.

RS232(2) Serial II Male connector

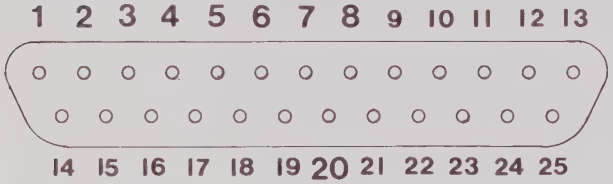
The RS232(2) port is a D-type male connector configured as DCE (Data Communication Equipment) which is able to accept a 25 way female connector (DTE).

It is generally used to connect the PPZ to a terminal, a tape reader, etc.

In 'local' or 'remote' mode it can be used to load data to the PPZ RAM or dump data from the PPZ RAM to peripheral equipment.

Only 9 of the 25 available pins play an active role in data transfer.

These are: numbers 1, 2, 3, 4, 5, 6, 7, 8 and 20.



The interior connectors of the male UART (Universal Asynchronous Receiver Transmitter) are configured in such a way as to make it identical (apart from ''pin 6 DSR'' or DCE to DCE connection) to the exterior connections of the Female Port.

This is done by inverting the function of individual pins. For instance: transmit data (pin Z) becomes received data (pin 3) and vice-versa.

Individual Pin Functions On The Serial II Connector

1.*	PG	PROTECTIVE GROUND
2.	TXD	RECEIVED DATA
3.	RXD	TRANSMITTED DATA
4.	RTS	DATA CARRIER DETECT
5.	CTS	DATA TERMINAL READY
6.	DSR	DATA SET READY
7.	SG	SIGNAL GROUND
8.	DCD	REQUEST TO SEND
20.	DTR	CLEAR TO SEND

*Pin number 1 is present in all connections, it represents the protective ground surrounding all the other cables.

8.2 CONNECTION TYPES

The PPZ supports the two most popular types of connection these are: XON/XOFF (3 wire cableform connection) and HARDWARE HANDSHAKE (7/8 wire cableform).

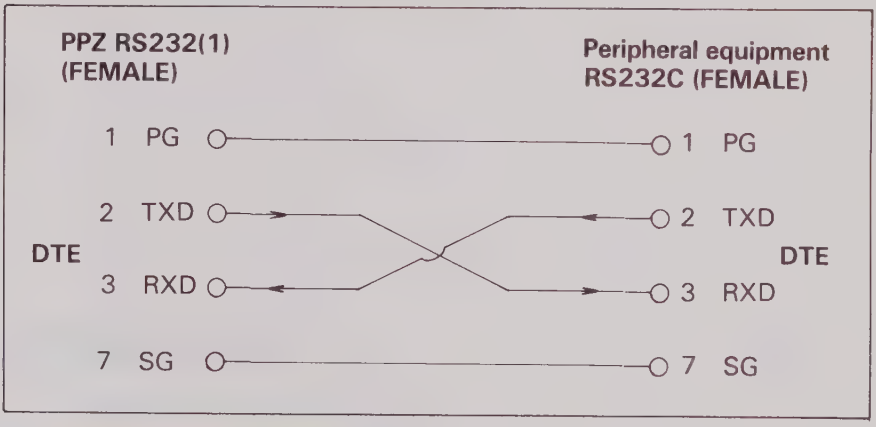
The most straightforward of these two is XON/XOFF.

XON/XOFF (3 wire cableform connection)

For connection of two alike machines a 'crossover' is needed.

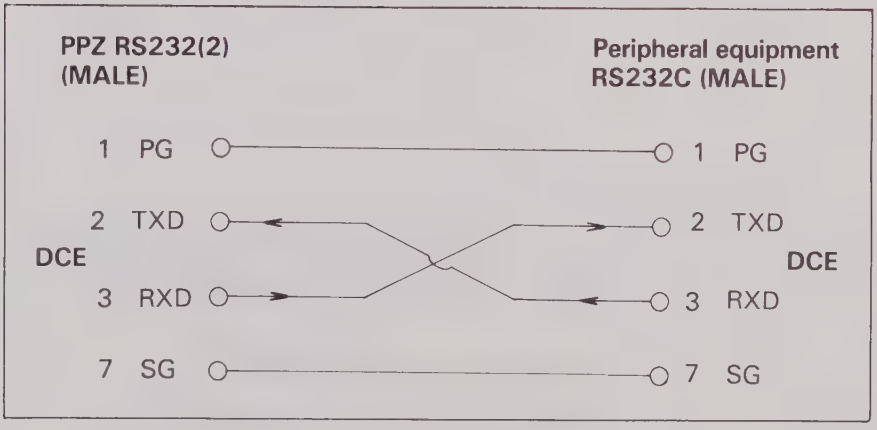
(a) DTE to DTE (crossover)

A 25-WAY MALE to MALE connector is required.



(b) DCE to DCE (crossover)

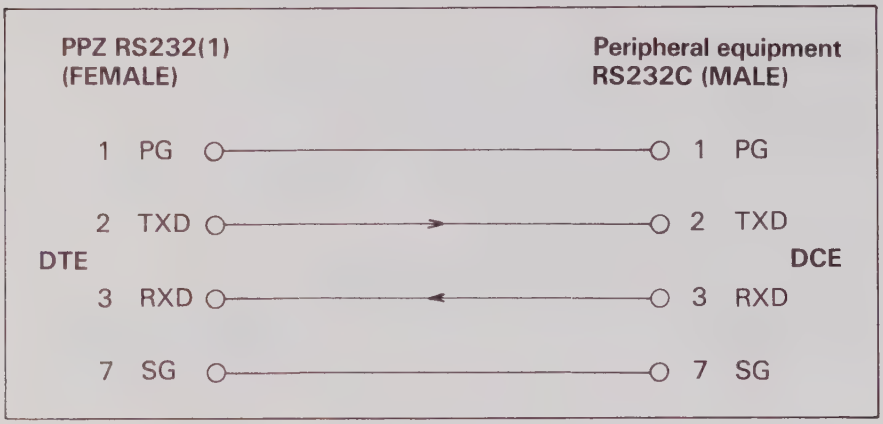
A 25-WAY FEMALE to FEMALE connector is required.



For connection of two unalike machines 'no' cross over is needed.

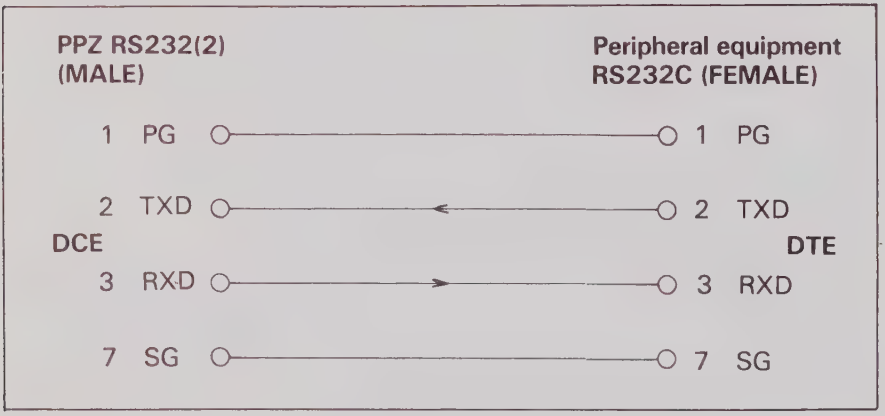
(c) DTE to DCE (straight)

A 25-WAY MALE to FEMALE connector is required.



(d) DCE to DTE (straight)

a 25-WAY FEMALE to MALE connector is required.



Note: Some machines do not have internal pull-ups and require extra connections within the cable form. Pull-ups may be required on pins 5, 6 and 8 of the external device if it is DTE or pins 4, 6 and 20 if it is DCE.

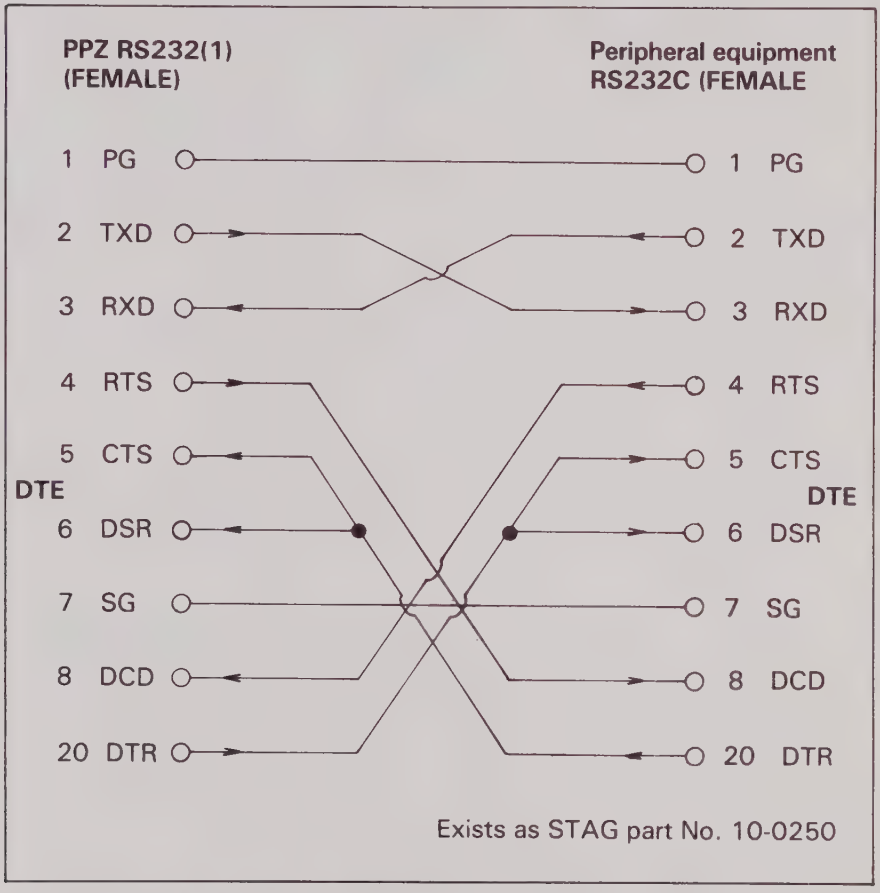
8.3 HARDWARE HANDSHAKE (7 OR 8 WIRE CABLEFORM)

DTE to DTE and DCE to DCE	Crossover	8 wire cableform
DTE to DCE and DCE to DTE	Straight	7 wire cableform

For connection of two alike machines a 'crossover' is needed.

(a) DTE to DTE (crossover)

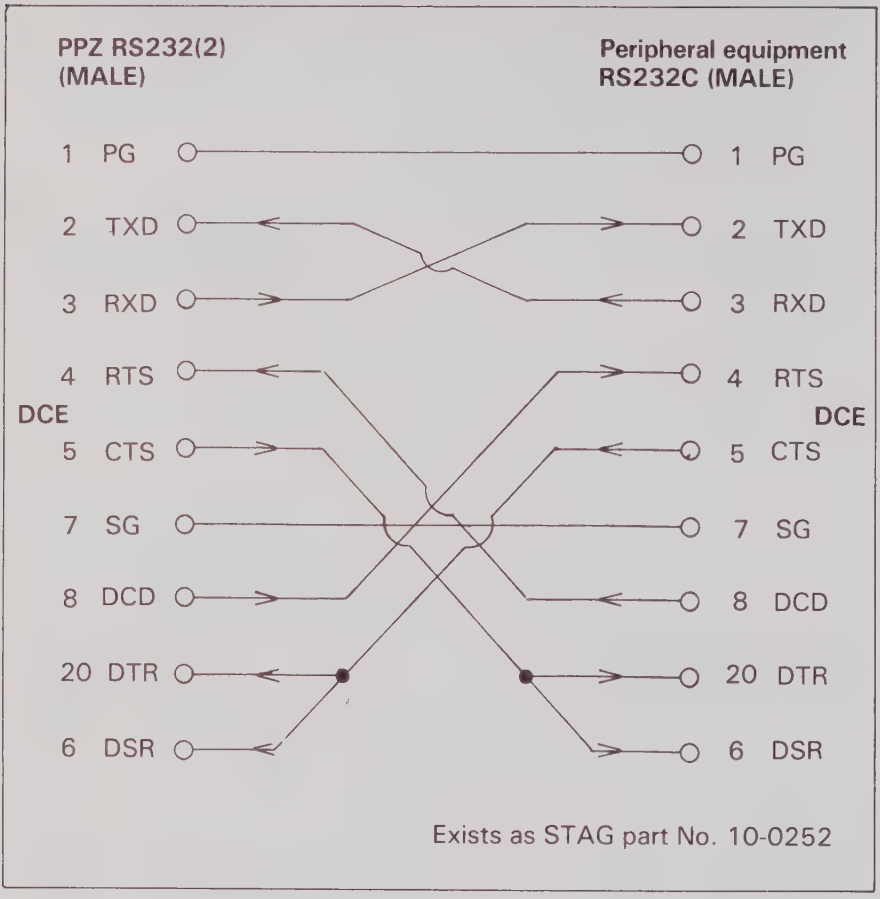
A 25-WAY MALE to MALE connector is required.



e.g. HP 300 Series
HP 98644 to RS232 I.

(b) DCE to DCE (crossover)

A 25-WAY FEMALE to FEMALE connector is required.

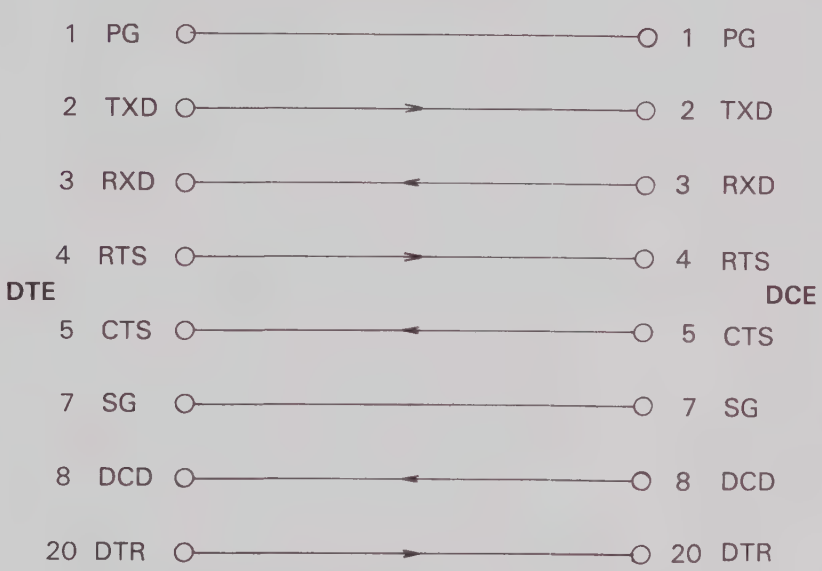


Exists as STAG part No. 10-0252

For connection of two unalike machines NO cross over is required.

(c) DTE to DCE (Straight)

A 25-WAY MALE to FEMALE connector is required.

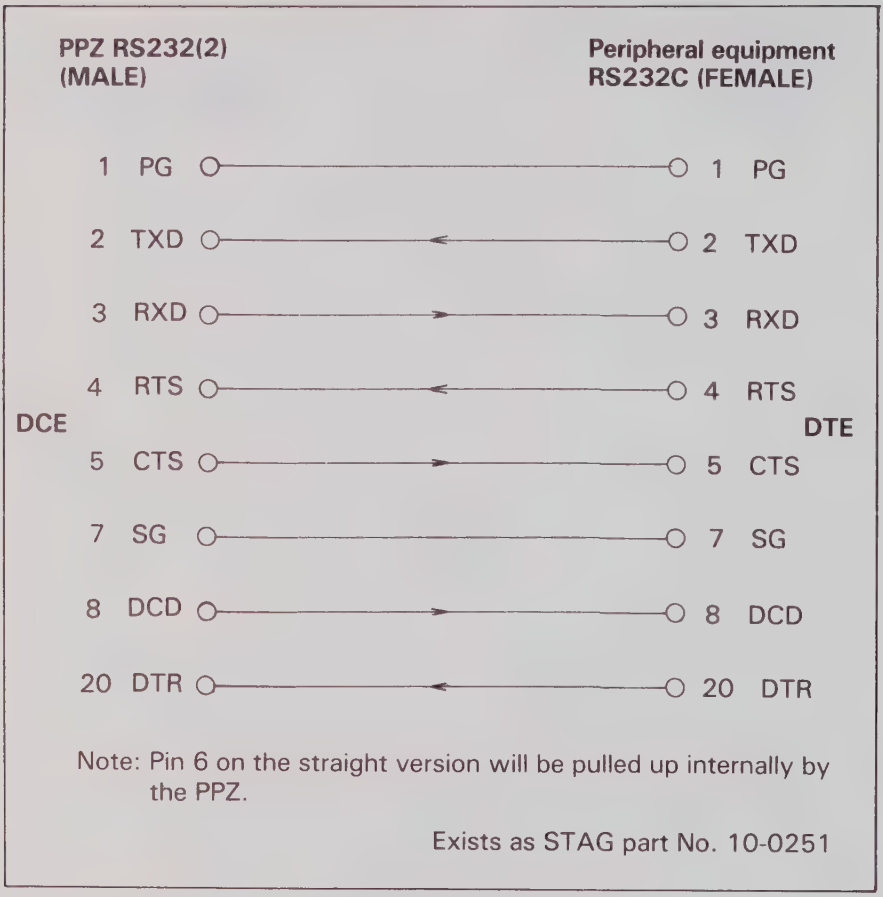


Note: Pin 6 on the straight version will be pulled up internally by the PPZ.

Exists as STAG part No. 10-0251

(d) DCE to DTE (Straight)

A 25-WAY MALE to FEMALE connector is required.



Note: Pin 6 on the straight version will be pulled up internally by the PPZ.

Exists as STAG part No. 10-0251

SECTION 9

stag

Sophisticated systems for the discerning engineer.

9.—REMOTE CONTROL
Via The RS232C
and the IEEE Ports

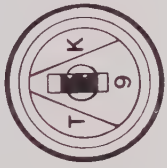
9.1 SELECTING THE OPERATING MODES

Selection of the operating mode is made by use of the three position key-switch located underneath the hinged flap. By inserting the supplied key one of these modes can be chosen.

LOCAL MODE (arrow vertical)

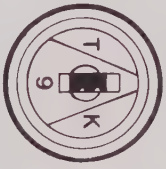
In LOCAL mode all functions of the PPZ are controlled from its own keyboard. All the functions operate interactively with the CRT and Input/Output transmissions take place between the PPZ and the peripheral equipment under 'Local Control'.

REMOTE MODE (arrow pointing to the left)



In REMOTE the PPZ operates under remote control from a computer or a terminal. The keyboard of the PPZ is inoperative at this time and the CRT will only display information as requested under remote command.

EDIT LOCK-OUT MODE (arrow pointing to the right)



The Edit Lock-out mode ensures all editing key functions are inoperative but that all other functions operate normally under LOCAL MODE conditions.

In this mode data corruption or changes to the RAM data cannot take place and security of information is ensured.

Interchange between LOCAL and REMOTE MODE can be accomplished by turning the key-switch to the required direction and pressing RESET.

Interchange between LOCAL and EDIT LOCK-OUT can be made by simply turning the key to the required direction. (Should an edit key function be requested under Edit Lock-Out Mode the CRT will display: EDIT LOCKED OUT).

9.2 SETTING UP PROCEDURE FOR REMOTE CONTROL

Before engaging REMOTE CONTROL, two selections have to be made:

1. the required serial interface.

2. the REMOTE CONTROL mode.

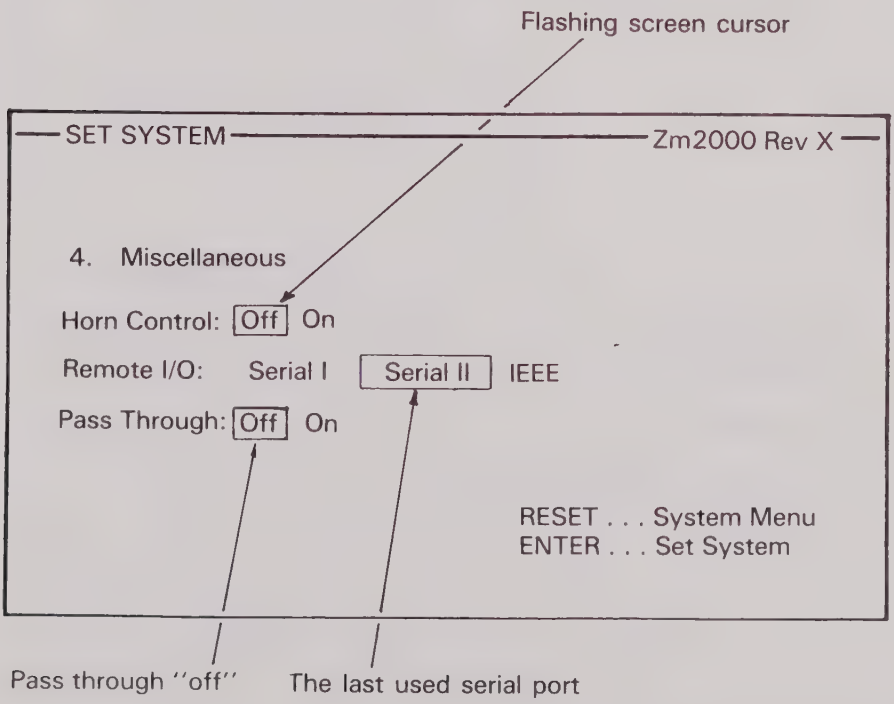
Selecting the required serial interface

From the SYSTEM MENU:

PRESS KEY A followed by KEY 4

This will display the 'Miscellaneous' page of the Set System option which will show a flashing screen cursor defaulted to HORN CONTROL and two static cursors. One of the static cursors will indicate the interface port last used in the REMOTE CONTROL mode and the other will indicate whether PASS-THROUGH IS 'on' or 'off'.

For example:

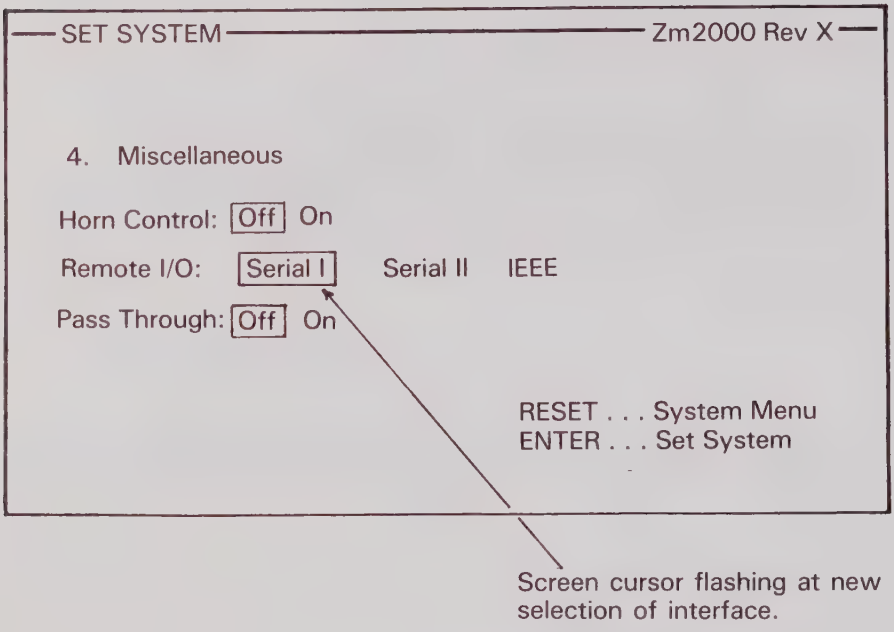

Flashing screen cursor
SET SYSTEM
Zm2000 Rev X
4. Miscellaneous
Horn Control: Off On
Remote I/O: Serial I Serial II IEEE
Pass Through: Off On
RESET . . . System Menu
ENTER . . . Set System
Pass through "off"
The last used serial port

If, as in this example the serial II port was previously used in REMOTE MODE and serial I is the new requirement, then it can be selected as follows:

PRESS KEY ↓

This will move the flashing cursor onto serial II. To select serial I:

PRESS KEY ←


SET SYSTEM
Zm2000 Rev X
4. Miscellaneous
Horn Control: Off On
Remote I/O: Serial I Serial II IEEE
Pass Through: Off On
RESET . . . System Menu
ENTER . . . Set System
Screen cursor flashing at new selection of interface.

To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting REMOTE CONTROL mode

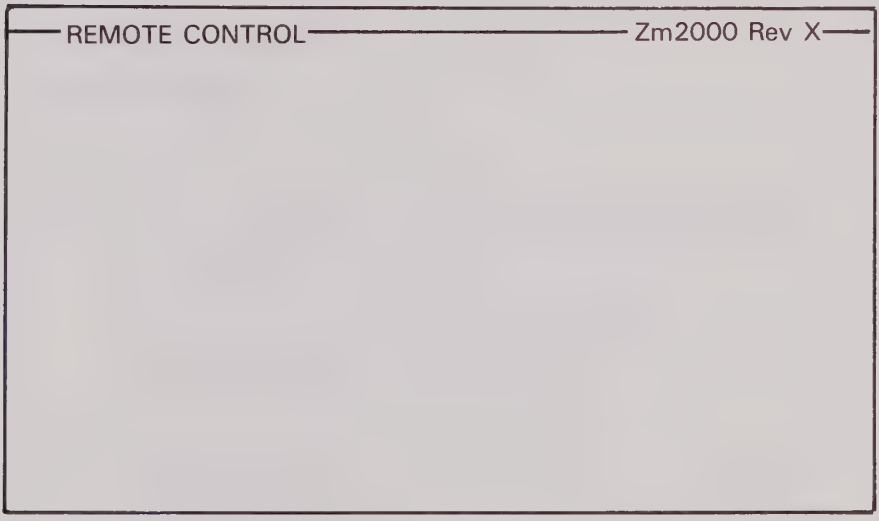
Selecting the REMOTE MODE involves the 3-position key switch which is located beneath the hinged flap on the top panel of the PPZ.

To select REMOTE MODE from LOCAL MODE:

TURN THE KEY CLOCKWISE (arrow pointing left)

PRESS RESET

The screen will display:



The programmer is now configured for operation in REMOTE CONTROL mode via either of the serial interface ports and the keyboard of the PPZ will be inoperative.

To return to the LOCAL mode:

TURN THE KEY COUNTER CLOCKWISE ONE STOP (arrow pointing down)

PRESS RESET

The SYSTEM MENU will now be displayed.

9.3 REMOTE CONTROL COMMANDS

RETURN Executes a command* for instance G RETURN, A6AF<RETURN, [RETURN, 11A RETURN etc.

G Software revision number. This command issues a 4-digit hex number representing the software configuration in the programmer.

Z Exits from remote control.

* Pressing RETURN for a second time will abort a command.

Set Up for Load and Program

Device type selection

hhhh = 4 hex digits

hhhh@ **Sets up the programmer for a particular device. The first two digits represent the manufacturer code and the second two represent the pin-out code.

[** Returns the four digit hex code of the device in use. The first two digits represent the manufacturer code and the second two digits represent the pin-out code.

T ILLEGAL-BIT CHECK: checks that there are no unprogrammable bits in the device.

B BLANK CHECK: checks that there are no programmed bits in the device.

R 'Respond', indicates device status for instance: 0FFF/8/0>. The first 4 digits reflect the working RAM limit relevant to the device.

the 5th digit is the byte size measured in bits and

the 6th digit reflects the unprogrammed state of the device selected. This digit can be either a 0 or a 1.

0 = Unprogrammed state FF.
1 = Unprogrammed state 00.

**See list of devices and device codes for the Zm2000 module

Device/RAM address limits

hhhh hhhh< Sets the lower address limit within the RAM.

hhhh hhhh; Sets the upper address limit within the device.

hhhh hhhh: Sets the lower address limit within the device.

L	LOAD: Loads device data into the RAM.
P	PROGRAM: Programs RAM data into the device.
V	VERIFY: Verifies device against the RAM.
S	CHECKSUM: Causes the programmer to calculate a checksum of RAM data.

8]22
16]22
32]22
RAM increment commands for 8, 16, 32 bit data.

nn↑ Fill RAM with any data byte, e.g. FF↑

By initiating either the LOAD or PROGRAM operation, data transference will commence between the RAM and the device inclusive of any pre-selected parameters specified above.

Set Up for Input and Output

Selection of translation formats

10A	BINARY	
11A	DEC BINARY	
50A	HEX-ASCII (Space)	All covered by standard HEX-ASCII
51A	HEX-ASCII (Percent)	
52A	HEX-ASCII (Apostrophe)	
53A	HEX-ASCII (Comma)	
59A	STAG HEX	
82A	MOTOROLA S-RECORD	
83A	INTELLEC 86	
86A	TEK HEX	
96A	EXTENDED TEK HEX	
81A	MOS TECHNOLOGY	

Input/Output Address Limits

hhhh hhhh< Sets a four digit figure defining the lower address limit.

hhhh hhhh; Sets the number of bytes of data to be transferred, which in effect defines the upper address limit.

hhhh hhhhW Sets the offset required for data transference for both INPUT and OUTPUT.

I INPUTS data from the computer/terminal to the PPZ RAM.

O OUTPUTS data from the PPZ to the computer/terminal.

By initiating either the INPUT or the OUTPUT operation data transfer will commence, inclusive of any pre-selected parameters specified above.

Error Responses

F — Error-status inquiry returns a 32-bit word that codes errors accumulated. The error-status word resets to zero after interrogation. (See PPZ remote error words; Section 9.3-06).

X — Error-code inquiry. The programmer will output error codes stored in the scratch-RAM and then clear them from memory. (See PPZ remote error codes: Section 9.3-05).

H — No operation. This is a null command and always returns a prompt character (>).

Programmer Responses

> RETURN — Prompt character. Informs the computer that the programmer has successfully executed a command.

F RETURN — Fail character. Informs the computer that the programmer has failed to execute the last entered command.

? RETURN — Question mark. Informs the computer that the programmer does not understand a command.

Remote Error Codes — 'X' — remote code PPZ

CODE	NAME	DESCRIPTION
20	Blank check Error	Device Not blank
21	Illegal bit Error	
22	Programming Error	The device selected could not be programmed
23	Verify Error	
26	Device Faulty	Either faulty part or reversed part
48	Buffer Overflow	
82	Checksum Error	
84	Invalid Data	

PPZ REMOTE ERROR WORD —F—

BIT NUMBER	RECEIVE ERRORS
31	If any error has occurred, this bit is set
30	Not used
29	Not used
28	Not used
27	Not used
26	Serial-overrun error (42)
25	Serial-framing error (41, 43)
24	Command-buffer overflow, i.e.> 18 characters (48)
	PROGRAMMING ERRORS
23	Any device-related error
22	Device appears faulty to the machines electronics (26)
21	L2 + L3 > Device
20	Not used
19	Device not blank (20)
18	Illegal bit (21)
17	Non verify (23)
16	Incomplete programming or invalid device (22)
	I/O ERRORS
15	If any I/O error has occurred, this bit is set
14	Not used
13	Not used
12	Not used
11	Checksum error (82)
10	Not used
9	Address error, i.e. > word limit
8	Data not hexadecimal where expected (84)
	RAM ERRORS
7	RAM—hardware error
6	Not used
5	L2 + L3 > RAM
4	Not used
3	Not used
2	No RAM or insufficient RAM resident
1	RAM write error, or program-memory failure
0	Not used

INTERPRETATION OF THE ERROR STATUS WORD

EXAMPLE: 80C80084

8 —The word contains error information

0 —No receive errors

C —(=8+4); 8 = Device related error
4 = Start line not set high

8 —Device is not blank

0 —No input errors

0 —No input errors

8 —RAM error

4 —Insufficient RAM resident

9.4 SETTING UP PROCEDURE FOR REMOTE CONTROL OPERATION VIA THE IEEE INTERFACE

Setting up for the IEEE is a three stage operation.

1. Selecting the IEEE interface port.
2. Selecting the address parameter for the IEEE.
3. Selecting REMOTE CONTROL mode.

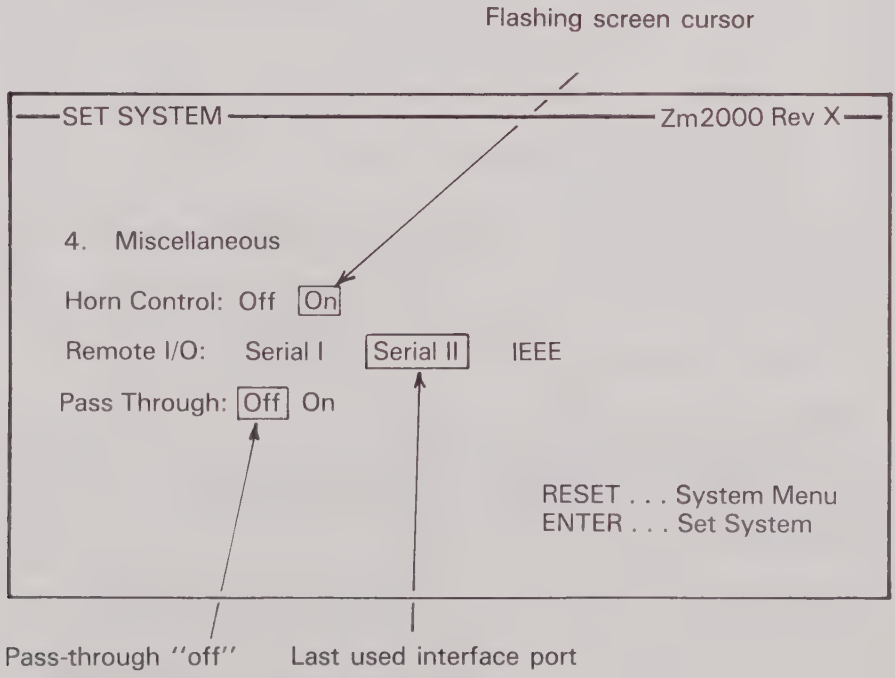
Selecting the IEEE interface port.

From the SYSTEM MENU:

PRESS KEY A followed by KEY 4

This will display the 'Miscellaneous' page of the Set System option which will show a flashing screen cursor defaulted to HORN CONTROL and two static cursors. One of the static cursors will indicate the interface port last used in the REMOTE CONTROL mode and the other will indicate whether PASS-THROUGH is 'on' or 'off'.

For example:

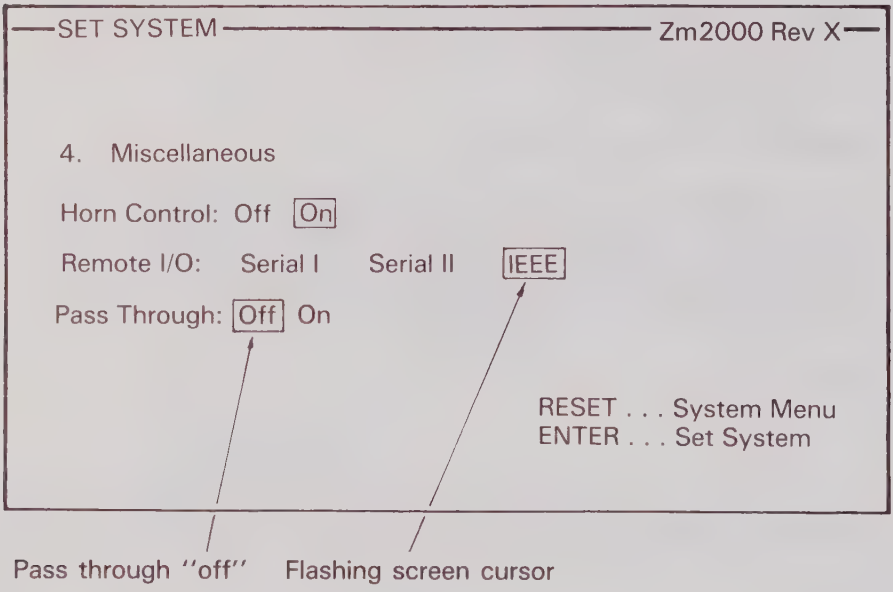


If, as in the example the Serial II port was previously used in REMOTE MODE, then to select the IEEE:

PRESS KEY ↓

This will move the flashing cursor onto Serial II. To select the IEEE:

PRESS KEY →


SET SYSTEM
Zm2000 Rev X
4. Miscellaneous
Horn Control: Off [On]
Remote I/O: Serial I Serial II [IEEE]
Pass Through: [Off] On
RESET . . . System Menu
ENTER . . . Set System
Pass through "off"
Flashing screen cursor

Note: The IEEE can only be selected with the PASS THROUGH turned "off".

To confirm this selection:

PRESS ENTER

To return the SYSTEM MENU:

PRESS RESET

Selecting the address parameter for the IEEE

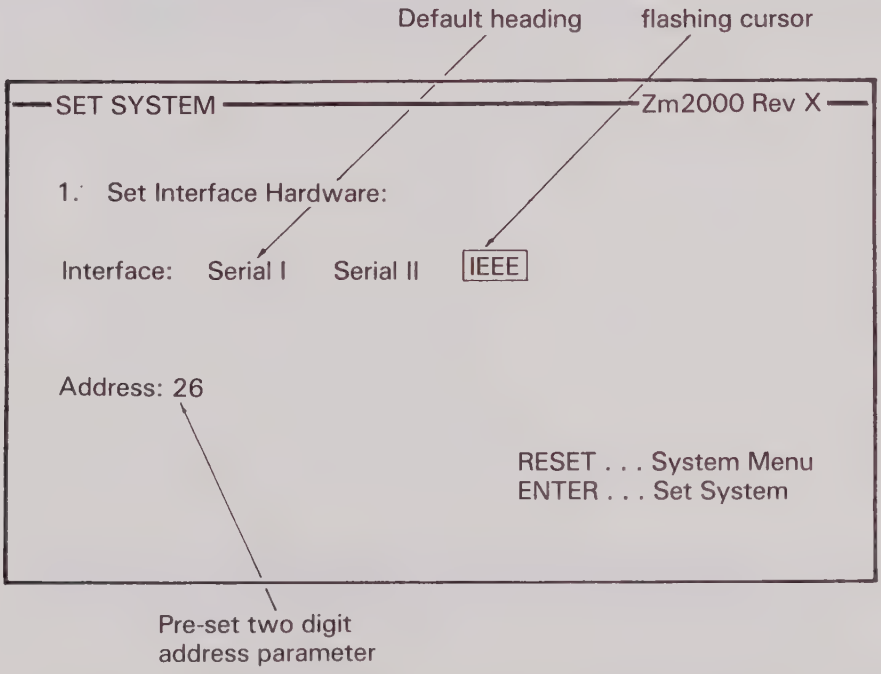
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the PPZ Programmer.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the Serial I heading. To select the IEEE:

PRESS KEY → twice



Pressing the down cursor key will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting Remote Control for the IEEE

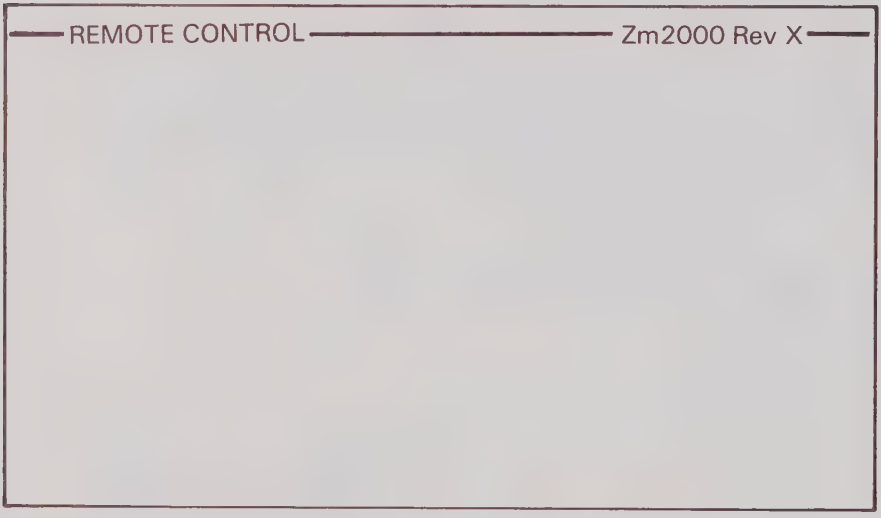
Selecting the REMOTE MODE involves the 3-position key switch which is located beneath the hinged flap on the top panel of the PPZ.

To select REMOTE MODE from LOCAL MODE:

TURN THE KEY CLOCKWISE (arrow pointing left)

PRESS RESET

The screen will display:



The programmer is now configured for operation in REMOTE CONTROL mode via the IEEE interface port and the keyboard of the PPZ will be inoperative.

To return to the LOCAL mode:

TURN THE KEY COUNTER CLOCKWISE ONE STOP (arrow pointing down)

ENTER COMMAND Z from the controller.

The SYSTEM MENU will now be displayed.

Structure of the IEEE as a Talker/Listener Port

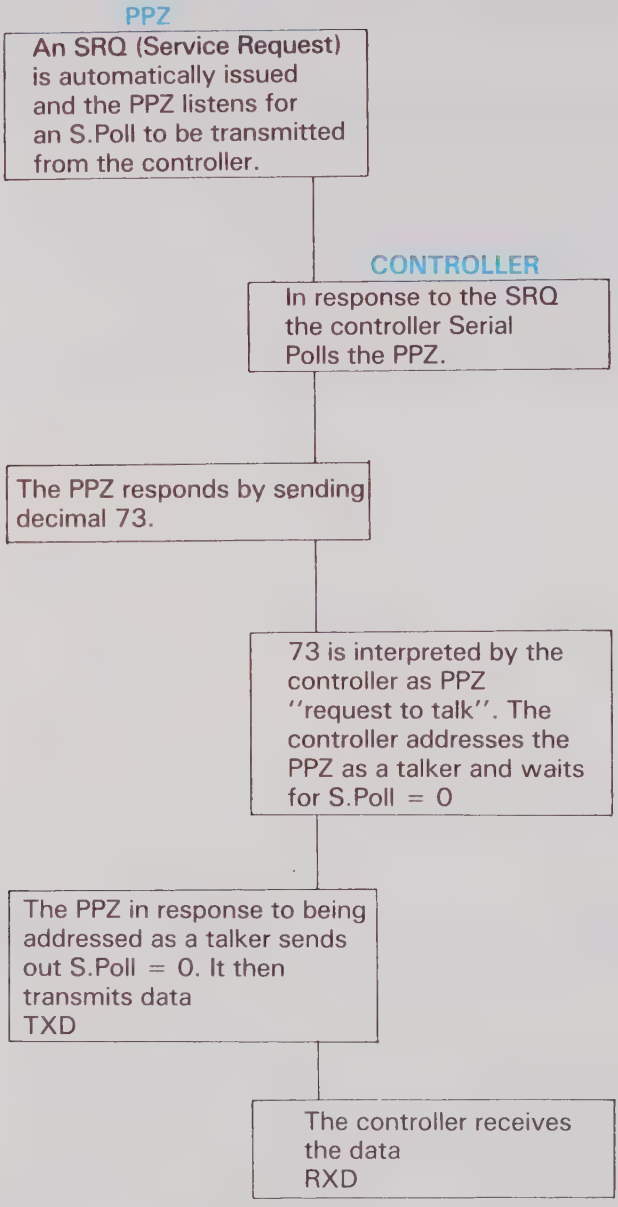
Certain communication procedures govern the set-up and operation of the 'PPZ' in the REMOTE CONTROL mode with a controller via the IEEE interface port.

Some of the communication is inherent to the controller and some is PPZ based but both are interlinked and will respond to interpretations of instructions generated by the other.

Note: The communication procedure must be followed precisely, in order to prevent ''bus lock-up'' (a state whereby communication between the two machines breaks down).

This procedure can be shown in the form of two flow charts.

(i) The PPZ as a talker (TXD)


PPZ
An SRQ (Service Request) is automatically issued and the PPZ listens for an S.Poll to be transmitted from the controller.
CONTROLLER
In response to the SRQ the controller Serial Polls the PPZ.
The PPZ responds by sending decimal 73.
73 is interpreted by the controller as PPZ "request to talk". The controller addresses the PPZ as a talker and waits for S.Poll = 0
The PPZ in response to being addressed as a talker sends out S.Poll = 0. It then transmits data TXD
The controller receives the data RXD

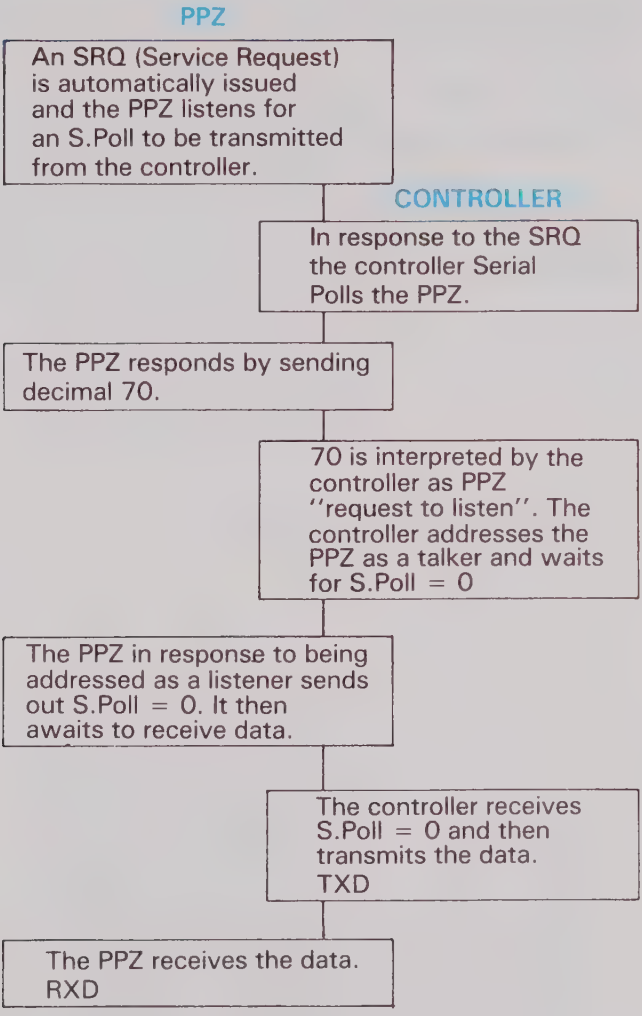
Notes:

(i) on power-up the PPZ transmits a pass response.

(ii) when the PPZ is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.

(iii) a rolling cursor will indicate a dump is in progress.

(ii) The PPZ as a listener (RXD)


PPZ
An SRQ (Service Request) is automatically issued and the PPZ listens for an S.Poll to be transmitted from the controller.
CONTROLLER
In response to the SRQ the controller Serial Polls the PPZ.
The PPZ responds by sending decimal 70.
70 is interpreted by the controller as PPZ "request to listen". The controller addresses the PPZ as a talker and waits for S.Poll = 0
The PPZ in response to being addressed as a listener sends out S.Poll = 0. It then awaits to receive data.
The controller receives S.Poll = 0 and then transmits the data. TXD
The PPZ receives the data. RXD

Notes:

(i) on power-up the PPZ transmits a pass response.

(ii) when the PPZ is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.

(iii) a rolling cursor will indicate a load is in progress.

IEEE 488 Interface port capabilities

Note: An abbreviated list of the IEEE capabilities is printed on the back panel below the port itself.

SH1 Full source handshake capability.

AH1 Full acceptor handshake capability.

T2 Full talker function capability and full serial Poll capability.

L2 Full listener function capability.

SR1 Full service request function capability.

RLO No remote/local function capability.

PP2 Parallel Poll capability without direct configuration from controller.

DCO No capability for device clear function.

DTO No capability for device trigger.

CO No controller function.

E1 Open collector drivers.

The following gives the signals on the IEEE 24 way connector:

Pin No	Name
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	DOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	GND

Pin No	Name
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	GND
19	GND
20	GND
21	GND
22	GND
23	GND
24	GND

Additional IEEE Remote Control Commands

The IEEE remote control commands are identical to those specified for the serial interface ports but some important additions should be noted.

(i) on power-up the PPZ has no capability (PPO) for P.Poll however, the programmer can be configured for PP2 (P.Poll capability) via the controller by entering ''NP]''. (N = single digit).

(ii) INPUT COMMAND FORMAT: EOI must be used to terminate command strings, thus CR, LF is NOT to be transmitted as a string terminator.

(iii) OUTPUT COMMAND FORMAT: EOI is sent with the last character (usually CR) of the response string.

SECTION 10

stag

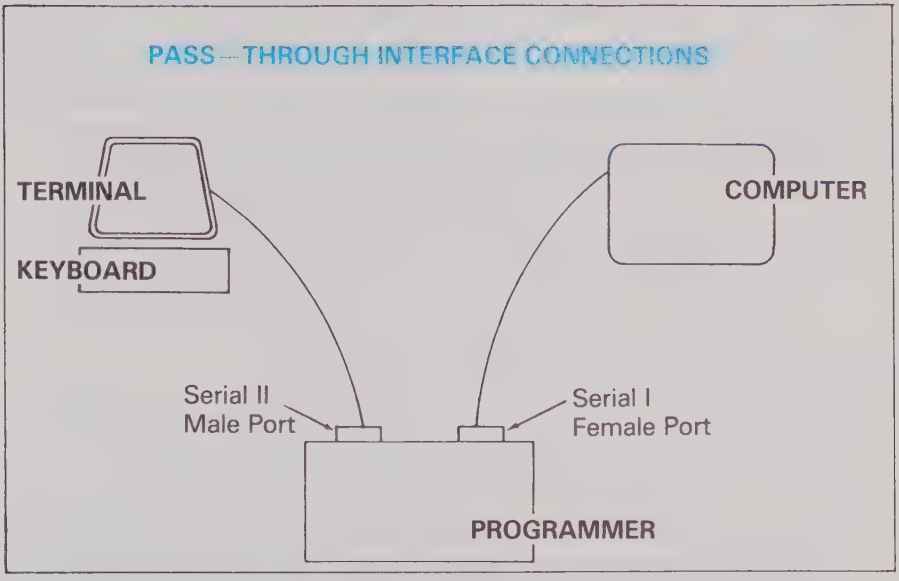
Sophisticated systems for the discerning engineer.

10.—PASS-THROUGH

10.1 PASS-THROUGH

The PASS-THROUGH mode allows communication between a terminal, the programmer and a computer by utilizing only one connection on the terminal and the computer.

Note: In order for PASS-THROUGH to work the computer must be connected to the programmer via the RS232(1) female port (configured as DTE) and the terminal must be connected via the RS232(2) male port (configured as DCE).



In PASS THROUGH the terminal has three modes of control. These are:

1. **The terminal controls the computer alone.**

2. **The terminal controls the PPZ via the computer.**

3. **The terminal can implement the INPUT and OUTPUT of data between itself and the PPZ or the computer and the PPZ.**

Selection of a particular mode is dependent upon the configuration entered on the PPZ.

1. The terminal can control the computer alone by selecting the system menu on the PPZ.
   When the ''system menu'' is entered on the PPZ the programmer will be ignored by both the terminal and computer, and will act merely as a channel for data transfer, hence ''pass-through''. (The 'system menu' is the default state of the PPZ when it is in LOCAL MODE).

2. The terminal can control the PPZ via the computer by selecting the remote control on the PPZ.

   By turning the keyswitch under the hinged flap to the ''left'' the REMOTE CONTROL can be entered on the PPZ.

   Terminal instructions can be passed through the programmer and sent to the computer. The computer can then motivate the programmer by returning the instructions to it.

   In effect the terminal will give the appearance of controlling the PPZ on a one to one basis, i.e.: REMOTE CONTROL, with all instructions identical to those specified in the REMOTE MODE.

3. The terminal can implement the input and output of data between itself and the PPZ or the computer and the PPZ by selecting the I/O mode on the PPZ*.

This can be shown by using an example. e.g. a data dump from the PPZ to the terminal.

To implement a data dump from the PPZ to the Terminal, OUTPUT via the Serial II interface must first be selected by entering into the PPZ I/O mode.

When these selections have been made pressing the ENTER key will put the PPZ into a state of readiness for the dump.

The necessary instruction to implement this dump is sent from the terminal through the programmer to the computer and back to the programmer.

This will cause the PPZ to 'dump' data (within preselected limits) back to the terminal along the same route.

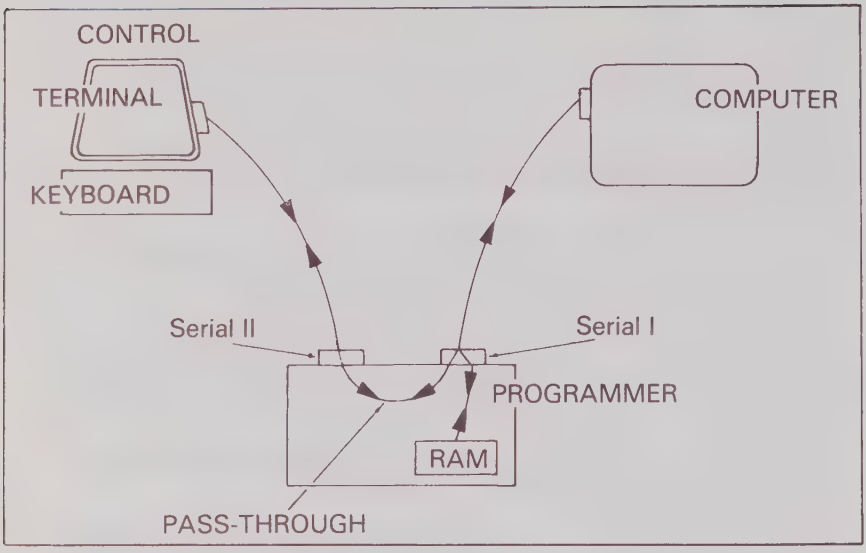
Therefore, selection of either load or dump between the programmer and the computer or the terminal must be made in the PPZ's I/O mode.

SELECTION OF INPUTS AND OUTPUTS IN THE PASS THROUGH MODE

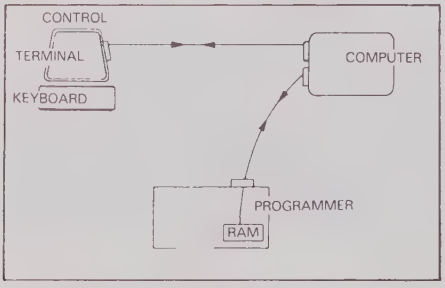
INPUT SERIAL I	Data is loaded from the computer to the PPZ
OUTPUT SERIAL I	Data is dumped to the computer from the PPZ
INPUT SERIAL II	Data is loaded from the terminal to the PPZ
OUTPUT SERIAL II	Data is dumped to the terminal from the PPZ

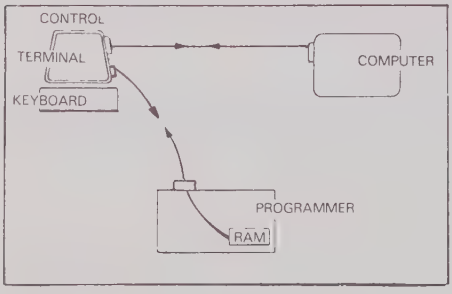
*If required address limits and offsets can also be entered in the PPZ's I/O mode.

A data file being dumped to the PPZ in PASS-THROUGH must be pre-fixed with a control—O (ASCII code—0F). This switches the data into the programmer RAM.



In effect PASS-THROUGH duplicates the operating structures as shown below.





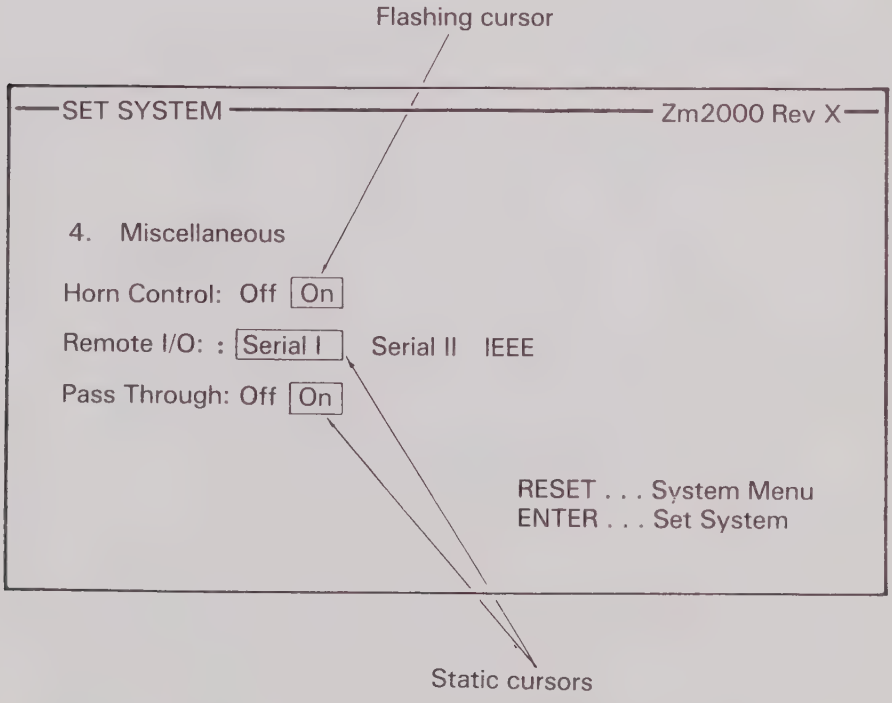
Selecting ''PASS-THROUGH''

To enter the ''PASS-THROUGH'' mode from the SYSTEM MENU:

PRESS KEY A followed by KEY 4

This will display the 'Miscellaneous' page of the Set System option which will show a flashing screen cursor defaulted to HORN CONTROL and two static cursors. The static cursors will indicate the last used mode of operation other than LOCAL. This could be either PASS-THROUGH itself or REMOTE MODE through one of the three displayed interface ports, ie: Serial I, Serial II or IEEE.

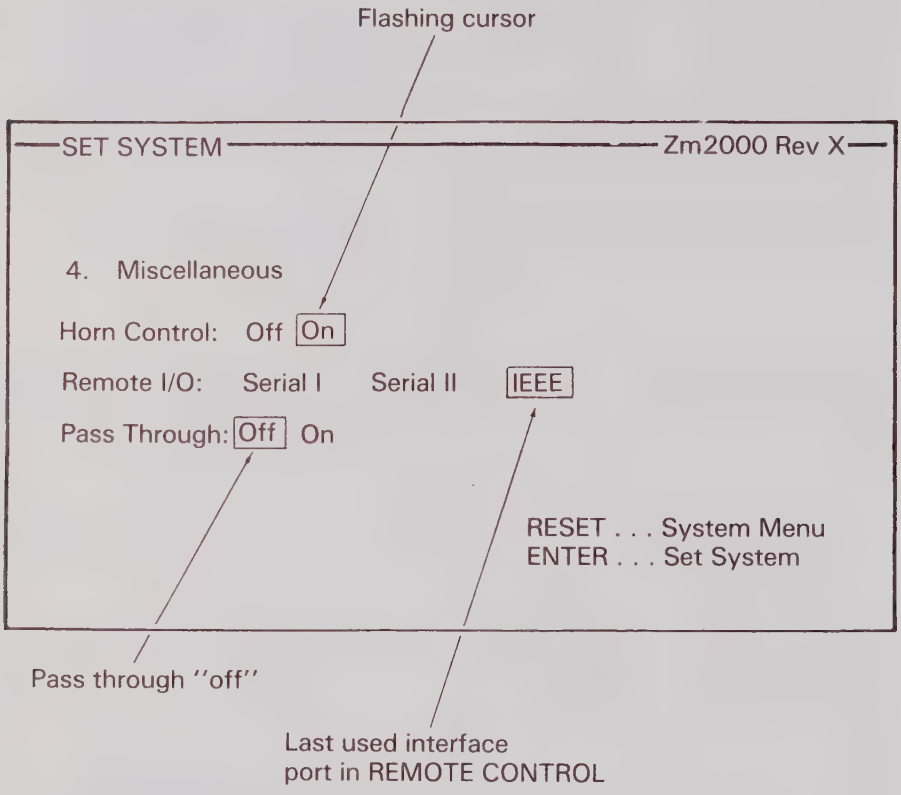
If the last used mode was PASS-THROUGH the display will show:



Note: PASS-THROUGH will always default to the Serial I port.

If the last used operating mode was REMOTE CONTROL the static cursors will show (i) the PASS-THROUGH ''off'' and (ii) the last interface port used in REMOTE CONTROL.

For example:



Selection of PASS-THROUGH can now be made by doing the following:

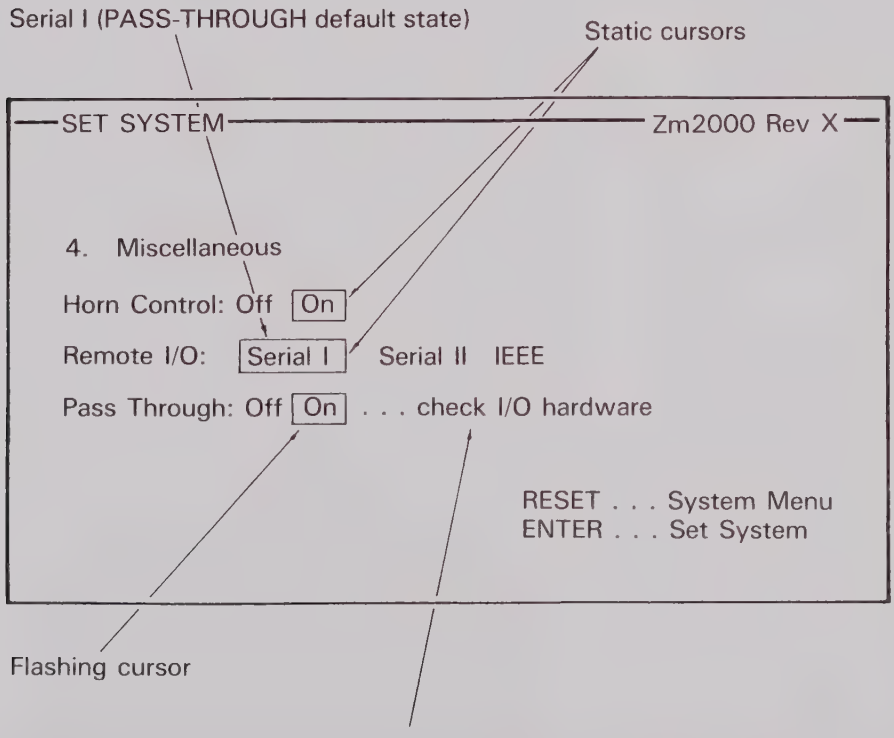
PRESS KEY ↓ twice

This will move the flashing cursor onto "PASS-THROUGH: Off". To select "Pass-Through: On":

PRESS KEY→

"Pass-Through: On" will now be indicated by the flashing screen cursor. Simultaneously the middle screen cursor will default to Serial I if it was previously indicating either of the other two ports.

For example:


Serial I (PASS-THROUGH default state)
Static cursors
SET SYSTEM
Zm2000 Rev X
4. Miscellaneous
Horn Control: Off On
Remote I/O: Serial I Serial II IEEE
Pass Through: Off On . . . check I/O hardware
RESET . . . System Menu
ENTER . . . Set System
Flashing cursor

The display will also show a visual reminder to check that the PPZ's 'I/O hardware' i.e. Baud rate, Parity, etc., correlate for both serial interface ports during PASS-THROUGH.

To check the 'I/O hardware':

PRESS ENTER followed by KEY 1

The PPZ is now configured for the PASS-THROUGH mode.

SECTION 11

stag

Sophisticated systems for the discerning engineer.

Zp006—LIGHT PEN OPERATING INSTRUCTIONS

Light Pen—Setting-Up Procedure

Z-Pack Installation

Install the Zp006 under the top cover of the PPZ, ensuring that the Stag emblem is facing upwards. Push the light pen into the keyed socket provided above the power switch.

Calibration

On power up, the PPZ's CRT will display the message 'Insert a Z-Module or a Z-Pack and press reset key'. Set the CRT to the required brilliance. Hold the light pen to any part of the screen and simultaneously press the 'L' key on the keyboard. A light pen offset will be displayed on the lower right of the CRT(---*---). Move the pen across the offset to focus on the central asterisk. An offset number is displayed during this operation, and when it stabilises, move the light pen down and away from the offset display. The light pen is now calibrated to the brilliance of the CRT and further calibration is not required unless the brilliance is changed. Turn off the power while installing the appropriate programming module and re-apply power to facilitate light pen operation.

Light Pen Operating Instructions

Introduction

The light pen, in conjunction with Zp006, can be used on Zm2000 modules with software revisions 18 onwards. Light pen operation takes the effort out of data entry and manipulation but does not differ radically in format from the keyboard operation. The user is therefore advised to refer to the Zm2000 manual for a full explanation of the various functions and their corresponding screen displays.

The basic procedure is the same for all functions:

i. Place the pen nib on the screen over the required character or option.

ii. Press until a cursor appears or the option begins to flash, then release. The screen will alter accordingly.

Note: If the Horn Control is set to 'on', pen contact is confirmed by a 'bleep'. Should the PPZ's display protection mechanism shut off the video to the CRT, simply press the light pen to any part of the screen and the display is restored.

Functions are described in the order in which they appear on the PPZ system menu. For the sake of brevity, where several instructions appear together, the 'press the pen on X' format will be shortened to 'press X'.

1 . . . Set Device (Section 2.1 in the manual)

Pressing the pen on 'Set Device' will call up the Select Manufacturer Screen. The pen is then used to locate the required manufacturer and device category by cursor movement in the same way as the left/right cursor keys do during the keyboard operations.

Press Enter

The select pinout screen is displayed and a choice is made as described above.

Press Enter or Reset

to return to the System Menu.

2 . . . Device Description (Section 2.2)

Pressing the pen on 'Device Description' will call up a description of the selected device. As with the keyboard method, no screen interaction is possible.

Press Reset

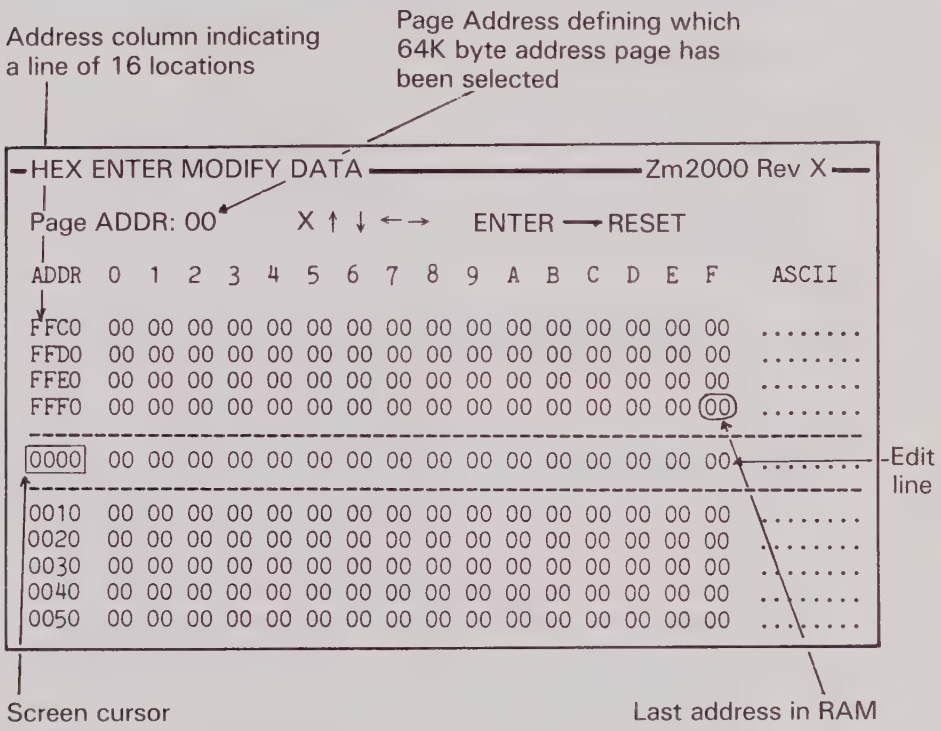
to return to the System Menu.

3 . . . Edit RAM (Section 5.2)

Pressing the pen on Edit RAM will call up the Edit RAM sub menu. Nine options are available, selected by number.

Hex, Decimal and Octal Editors

The three editors operate identically, and the pen can be used to change between them by pressing X.



With the cursor located in the 'ADDR' column,

Press ↓ or ↑

to increment or decrement the locations displayed by 8, 10 or 16.

To locate a specific location more quickly, press the pen on the required characters from 0 to F, then press Enter. All the addresses will adjust accordingly.

It is possible to switch between the page address and the specific location, using the pen to move the cursor, but the 64K page address can only be altered if the PPZ has RAM extension.

There are three methods for moving along the data line of the highlighted address:

i. Using the pen to move the cursor.

ii. Pressing the pen on Enter.
This causes the cursor to move one address to the right. Holding the pen on Enter will cause the cursor to move continuously to the right. A new line of data is moved into the edit line each time the cursor reaches the extreme right. Continuous contact will implement auto repeat.

iii. Pressing the pen on → or ←

→ Has the same effect as Enter
← Has the same effect as Enter, but cursor movement is to the left.

The data at a particular location is changed by using the pen to select the required characters from 0 to F.

Press —

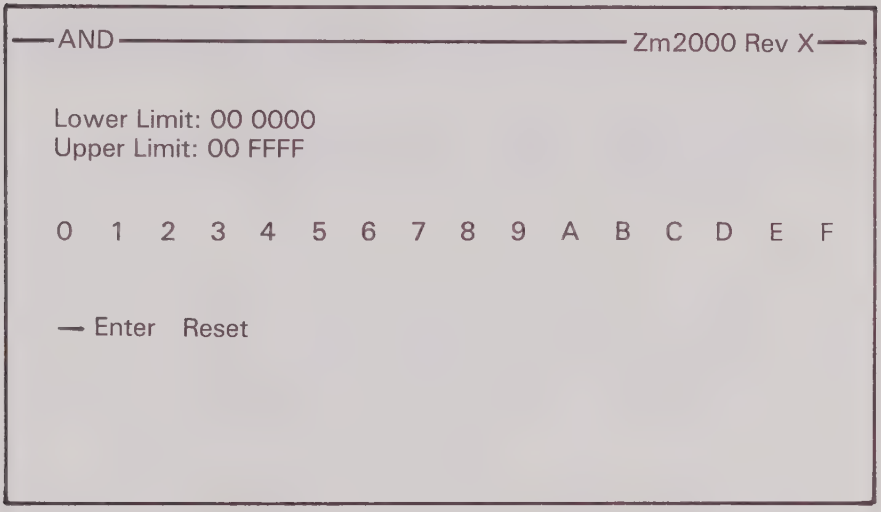
to return to the Edit RAM sub-menu

Press Reset

to return to the system menu.

Logical OR/Logical AND (Section 5.2)

The logical OR and AND functions are identical in operation.



The pen is used to move the cursor between address fields.

To alter the address limits use the pen to select the required characters from the pseudo keypad at the bottom of the screen.

Press Enter

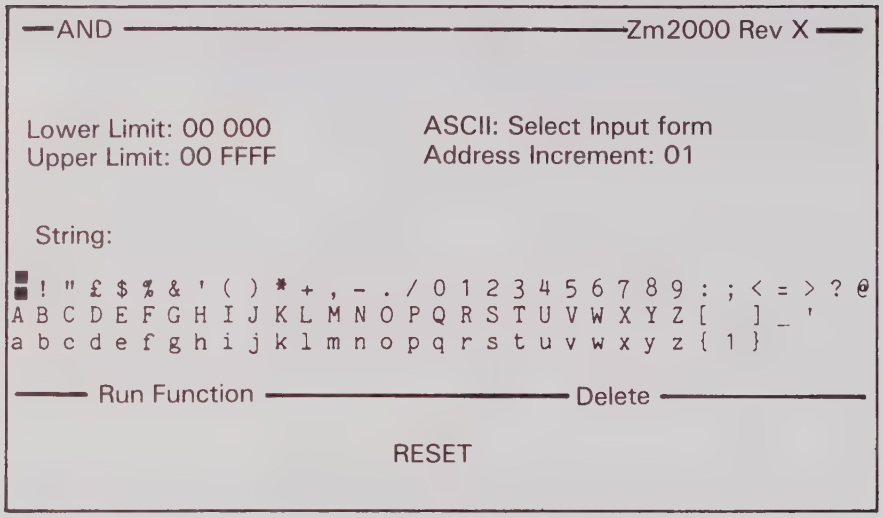
to display 'Address Increment'. This can also be changed using the pseudo keypad.

Press Enter

to display the choice of ASCII or numeric.

ASCII

If the pen is used to select ASCII, the display will show:


—AND
Zm2000 Rev X —
Lower Limit: 00 000
Upper Limit: 00 FFFF
ASCII: Select Input form
Address Increment: 01
String:
! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { 1 }
Run Function
Delete
RESET

The required string is formed by light pen selection of various characters.

Press Delete

to delete the last character selected

Press Run Function

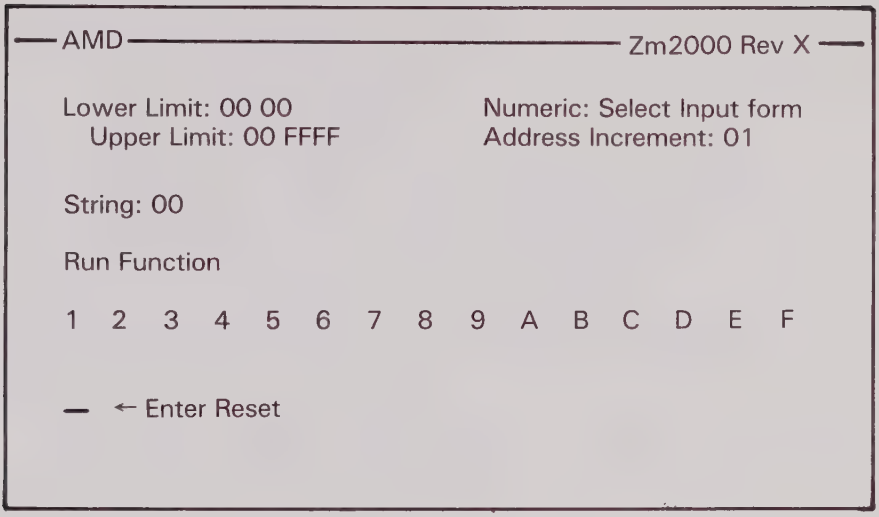
to implement the ASCII logical AND

Press —

to return to the 'Edit RAM Data' sub-menu.

Numeric

If the pen is used to select Numeric, the display will show:


AMD
Zm2000 Rev X
Lower Limit: 00 00
Upper Limit: 00 FFFF
Numeric: Select Input form
Address Increment: 01
String: 00
Run Function
1 2 3 4 5 6 7 8 9 A B C D E F
— ← Enter Reset

The required string is formed by light pen selection from the pseudo keypad, and the string can be lengthened by pressing Enter.

Press ←

to reduce the string size by the last entered byte.

Press Run Function

to implement the Numeric Logical AND

Press —

to return to the 'Edit RAM Data' sub-menu.

Nibble Swap (Section 5.2)

Operation is the same as for the logical AND/OR function but there is no ASCII or numeric option.

String Search (Section 5.2)

Operation is the same as for the logical AND/OR function, but there is no choice of Address Increment.

When Run Function is pressed, either the display will show the message 'not found', with the option of adjusting the specification or it will give the first location at which the string was found.

Pressing the pen on 'E: view in the editor' enables the user to see the located string in the RAM, and implement further editing if required.

Press —

to return to string search mode. Further string search operations may be performed.

Block Move (Section 5.2)

The operation is the same as for Nibble Swap except that Destination Address selection follows instead of Address Increment.

Complement (Section 5.2)

The operation is identical to that of Nibble Swap.

4 . . . Fill Ram (Section 5.3)

Pressing the pen on Fill RAM will call up the Fill RAM sub menu. Four options are available:

0 . . . Fill: 00, F . . . Fill: FF, E . . . Fill: Empty State

The operation of these three functions is identical to that of Nibble Swap.

A . . . Fill: Arbitrarily

The operation of this function is identical to that of logical AND/OR.

5 . . . List RAM (Section 5.1)

Operation is similar to that of Edit RAM. However, it is not possible to alter the data at a particular location, and therefore neither →← nor Enter can be used to move along the data line.

6 . . . List Device (Section 4.9)

Operation is similar to that of List RAM, but the device can only be listed in hexadecimal mode. It is not possible to enter a new page address as devices greater than 64K are not yet supported.

8 and 9. Checksum and CRC (Sections 4.7 and 4.8)

The operation of these functions is identical to that of Nibble Swap.

When the parameters have been set, pressing Enter will cause the checksum or CRC to be displayed, but there is still the opportunity to adjust the specification and thus obtain another checksum or CRC.

Press — or Reset

to return to the system menu.

A . . . Set System

Pressing the pen on A . . . Set System will call up the Set System sub menu. Four options are available:

Set Interface Hardware (Section 6.1)

The pen moves the cursor to select the required interface.

If the IEEE is chosen, the displayed parameters are removed and 'Address' replaces them. It can be changed by using the pen to select a new value from the pseudo keypad.

I/O Software (Section 6.1)

The pen moves the cursor to select the required format and control Z option.

Program/Test Control (Section 4.4)

The pen moves the cursor to choose the 'none', 'empty' or 'illegal bit' option, and alters the bit mask value by selection from the pseudo keypad.

Miscellaneous (Sections 9.2, 1.5, 10.1)

The pen moves the cursor to choose the remote I/O options, and whether the Horn Control and Pass-Through are on or off. If Pass-Through is on, the I/O option defaults to Serial I, and a message reminds the user to check the I/O hardware parameters.

Load (Section 4.1)

Pressing the pen on load will call up the load display. The addresses and address increment can be changed by using the pen to select new values from the pseudo keypad.

PRESS ENTER

'Device loaded' and a checksum are displayed. A failure is displayed as for the keyboard method, and the pen is pressed on ↓ to see the next failing addresses or devices.

I/O (Section 6.2)

Pressing the pen on I/O will call up the I/O display. The offset address and RAM address limits can be changed by using the pen to select the new values from the pseudo keypad. I/O type and I/O interfaces are selected by using the pen to move the cursor.

Press Enter

to initiate the load or dump.

If the user wishes to interrupt the operation while it is in progress, it is necessary to press the MANUAL Reset Key.

Test (Section 4.3)

Pressing the pen on Test will call up the test display. The pen moves the cursor to choose 'Empty' or 'verify'. The bit mask option is only available for 'Verify'. The addresses and address increment can be changed by using the pen to select new values from the pseudo keypad.

Press Enter

'Device Verified'/'Device Empty' and a checksum are displayed. A failure is displayed as for the keyboard method and the pen is pressed on ↓ to see the next failing addresses or devices.

Program (Section 4.4)

Pressing the pen on Prog will call up the Program display. Operation is as for load, except that there is a choice of pre-program checks: None, Empty or Illegal Bit. To move between them it is necessary to press A . . Set System followed by 3 . . . Program/Test Control.

Note: To abort any of the above functions while they are in progress it is necessary to hit the MANUAL Reset Key.

Light Pen Operations using the Edit Lockout Option

The Light pen operates in this mode, but an attempt to edit will cause the message 'Edit Lockout Press Reset' to be displayed. Press the pen on Reset to return to normal.

Light Pen Operation using Remote Control

In order to exit from Remote Control it is necessary to press the MANUAL Reset Key.

SECTION 12

stag

Sophisticated systems for the discerning engineer.

Zp005—HANDLER PACK

OPERATING INSTRUCTIONS

(Zm2000 Module)

Mar 81 Rev 1 803 1/19

ZP005—CONTENTS

H.P. 1 Setting Up Procedure

H.P. 2 Test Mode

H.P. 3 Program Mode

H.P. 4 Error Messages

H.P.1 **Setting Up Procedure**

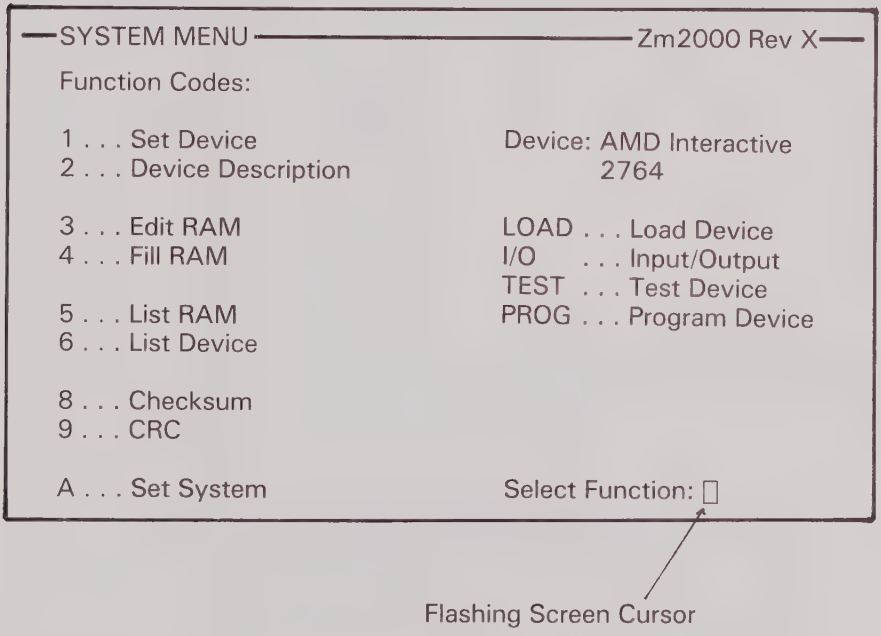
With the PPZ mainframe and the handler powered down:

1. Insert the Z-pack into the rack under the top cover of the PPZ ensuring that the Stag logo is facing upwards.

2. Plug-in the handler-lead to the handler socket (parallel 1) on the rear panel of the PPZ.

3. Power-up the PPZ and the handler. The standard System Menu will be returned with the Zm2000 Module.

For example:


SYSTEM MENU
Zm2000 Rev X
Function Codes:
1 . . . Set Device
2 . . . Device Description
3 . . . Edit RAM
4 . . . Fill RAM
5 . . . List RAM
6 . . . List Device
8 . . . Checksum
9 . . . CRC
A . . . Set System
Device: AMD Interactive 2764
LOAD . . . Load Device
I/O . . . Input/Output
TEST . . . Test Device
PROG . . . Program Device
Select Function:
Flashing Screen Cursor

If the handler lead type is incompatible with the handler pack a message ''Cable Error'' will appear instead of the handler message in Test and Program modes.
(See Section H.P. 4 Error Messages).

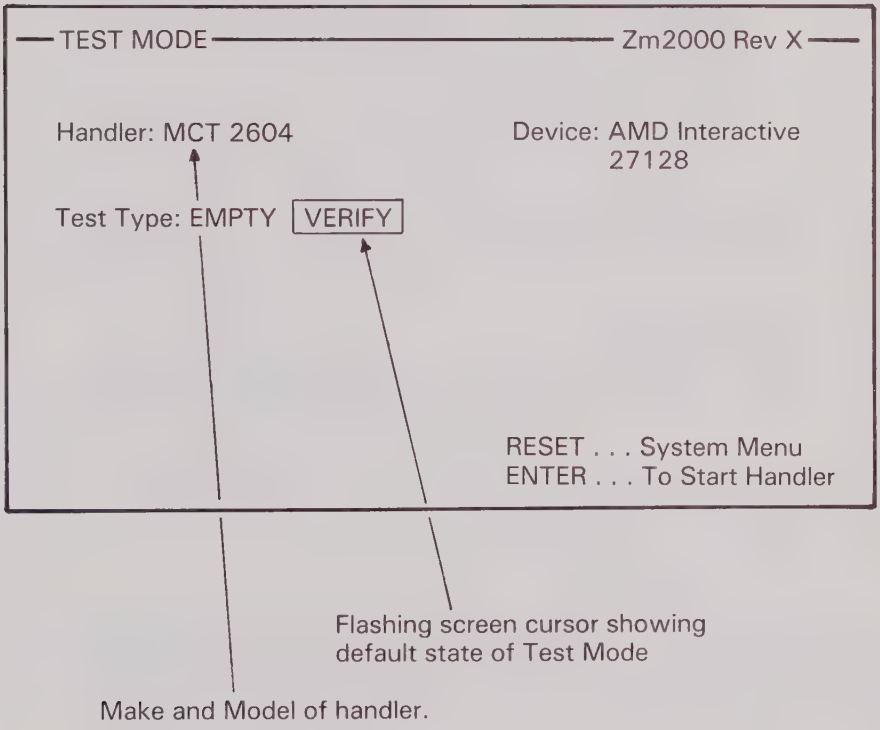
H.P.2 **Test Mode**

The handler pack allows devices in the handler to undergo tests for either the empty state or for verification of data.

To select Test Mode:

Press TEST

This will display:


TEST MODE
Zm2000 Rev X
Handler: MCT 2604
Device: AMD Interactive 27128
Test Type: EMPTY
VERIFY
RESET . . . System Menu
ENTER . . . To Start Handler
Flashing screen cursor showing default state of Test Mode
Make and Model of handler.

A screen cursor will be flashing on 'Verify' which is the default state of the Test Mode.

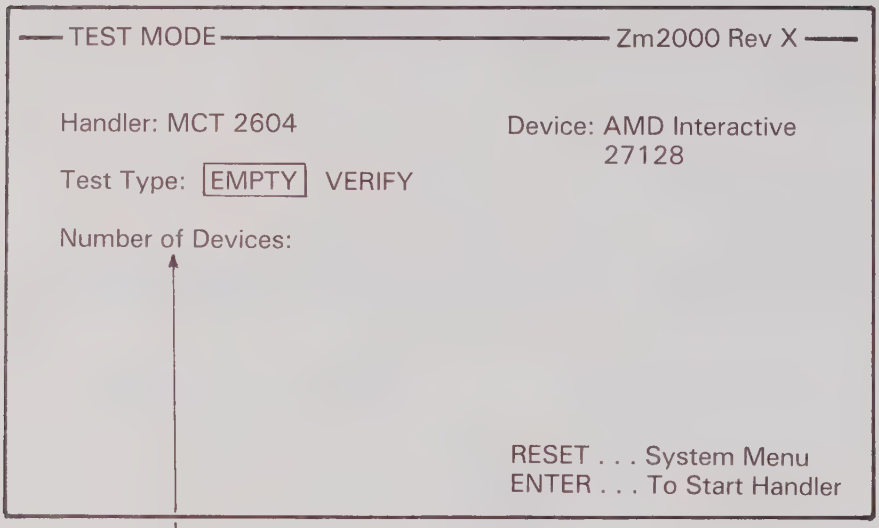
To select the 'Empty' test:

Press ←

To confirm the selection:

Press ENTER.

The screen will now display the following:


TEST MODE
Zm2000 Rev X
Handler: MCT 2604
Device: AMD Interactive 27128
Test Type: EMPTY VERIFY
Number of Devices:
RESET . . . System Menu
ENTER . . . To Start Handler

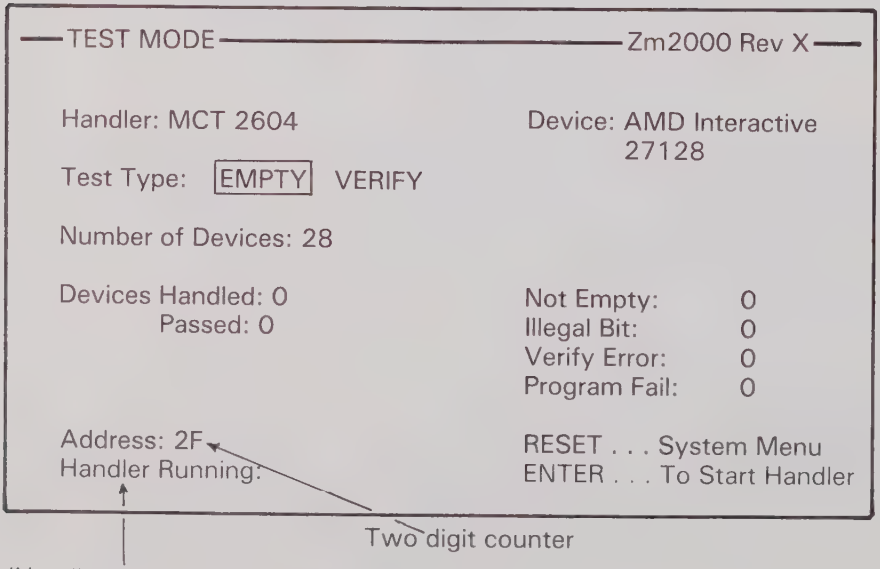
Flashing Screen Prompt

The number of devices required to pass 'Test' can now be entered from the keyboard. This can take values from 1 to 9,999.

To confirm the number of devices and initiate the Test function:

Press ENTER

The screen will display:



Two digit counter

'Handler Running' message will be shown during correct handler operation.

During the Test sequence a two digit counter will increment. At the end of a successful Test run the message 'End of Run' will be displayed in place of the 'Handler Running' message and a yield analysis will be produced.

The yield analysis will indicate the following:

(i)	The number of devices handled	— **Devices Handled**
(ii)	The number of devices passed	— **Devices Passed**
(iii)	The number of devices not empty	— **Not Empty**
(iv)	The number of devices with illegal bits	— **Illegal Bit**
(v)	The number of devices that do not verify	— **Verify Error**
(vi)	The number of devices not programmed	— **Program Fail**

In the following example 32 devices have been handled, 28 of which have passed and 4 have been found to be 'not empty':

```
── TEST MODE ──────────────────────── Zm2000 Rev X ──

  Handler: MCT 2604                 Device: AMD Interactive
                                            27128
  Test Type: [EMPTY] VERIFY

  Number of Devices: 28

  Devices Handled: 32               Not Empty:      4
        Passed: 28                  Illegal Bit:    0
                                    Verify Error:   0
                                    Program Fail:   0

  Address: 3F                       RESET . . . System Menu
  End of Run:                       ENTER . . . To Start Handler
```

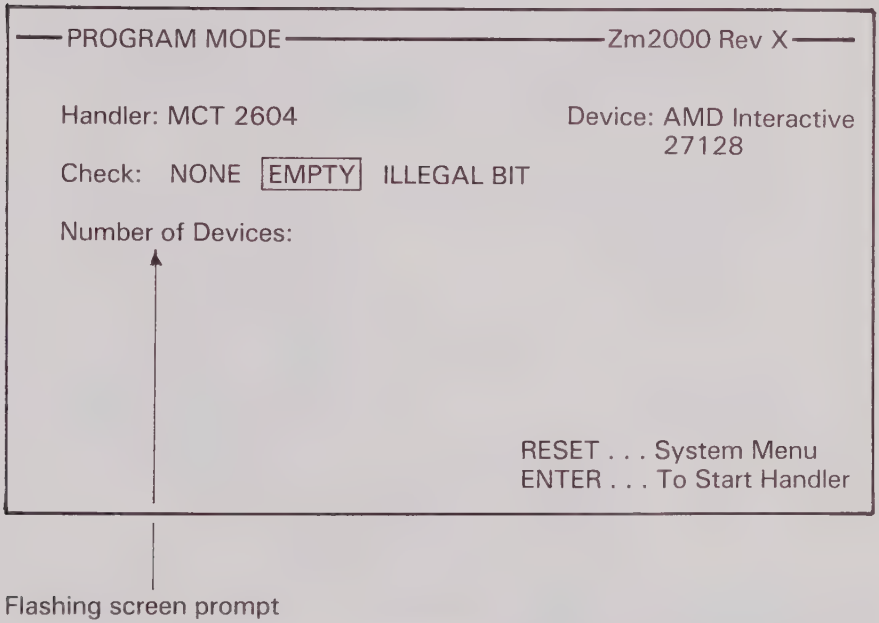
H.P.3 **Program Mode**

A maximum of 9,999 devices can be specified for correct programming in any one run using the Handler-Pack. All of the usual pre-program checks are carried out as well as the post-program verify.

To select Program mode:

Press PROG

This will display:


PROGRAM MODE
Zm2000 Rev X
Handler: MCT 2604
Device: AMD Interactive 27128
Check: NONE EMPTY ILLEGAL BIT
Number of Devices:
RESET . . . System Menu
ENTER . . . To Start Handler
Flashing screen prompt

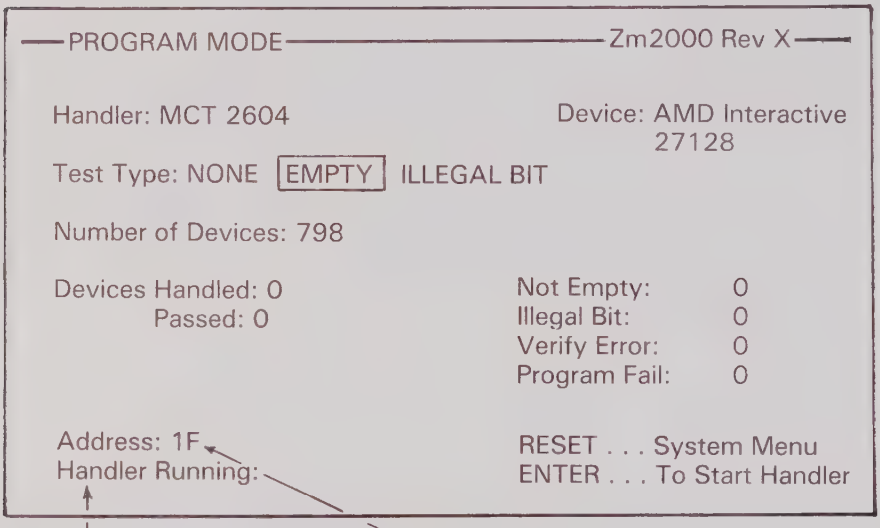
The pre-program checks can be selected in the usual way from the Set System option prior to programming.

The number of devices required to be programmed can now be entered as indicated by the flashing screen prompt. This value can be from 1 to 9,999 and is entered directly from the keyboard.

To confirm the number of devices and start the program run:

Press ENTER

The screen will display:



Two digit counter

'Handler Running' message will be shown during correct handler operation.

During the programming run a two digit counter will increment as each device is processed. At the end of a successful run the message 'End of Run' will be displayed in place of the 'Handler Running' message and a yield analysis will be produced.

The yield analysis will indicate the following:

(i)	The number of devices handled	— **Devices Handled**
(ii)	The number of devices passed	— **Devices Passed**
(iii)	The number of devices not empty	— **Not Empty**
(iv)	The number of devices with illegal bits	— **Illegal Bit**
(v)	The number of devices that do not verify	— **Verify Error**
(vi)	The number of devices not programmed	— **Program Fail**

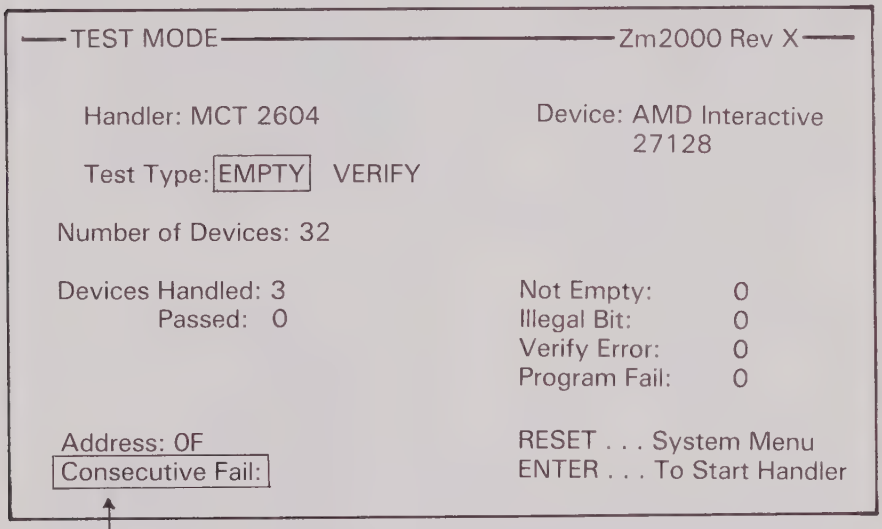
In the following example 800 devices have been handled, 798 of which have been programmed and 2 have not been programmed:

```
──PROGRAM MODE──────────────────Zm2000 Rev X──

Handler: MCT 2604                Device: AMD Interactive
                                         27128
Test Type: NONE [EMPTY] ILLEGAL BIT

Number of Devices: 798

Devices Handled: 800             Not Empty:     0
        Passed: 798              Illegal Bit:   0
                                 Verify Error:  0
                                 Program Fail:  2

Address: 3F                      RESET . . . System Menu
Handler Running:                 ENTER . . . To Start Handler
```

(i) Consecutive Fail

If 3 devices should fail any of the tests consecutively the test/program run will abort and the message 'Consecutive Fail' will appear in the bottom left hand corner.

For example:

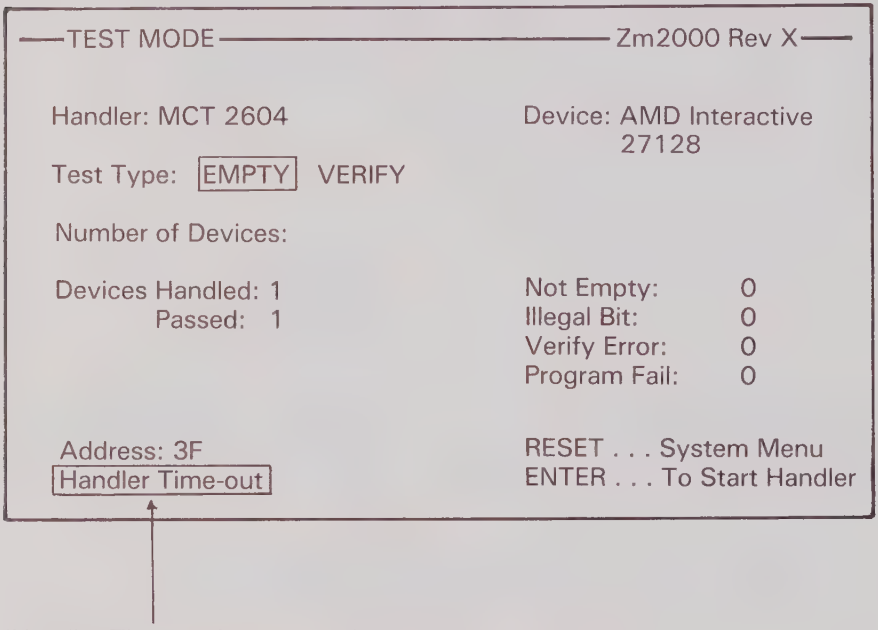


'Consecutive Fail' error message displayed in reverse video.

(ii) **Handler Time-out**

As each device is handled the PPZ waits for a response from the handler before allowing the next device to be tested or programmed. Should this response not be forthcoming due to a handler malfunction or connection failure then the test/program sequence will stop and the message 'Handler Time-out' will appear in the bottom left hand corner of the screen.

For example:

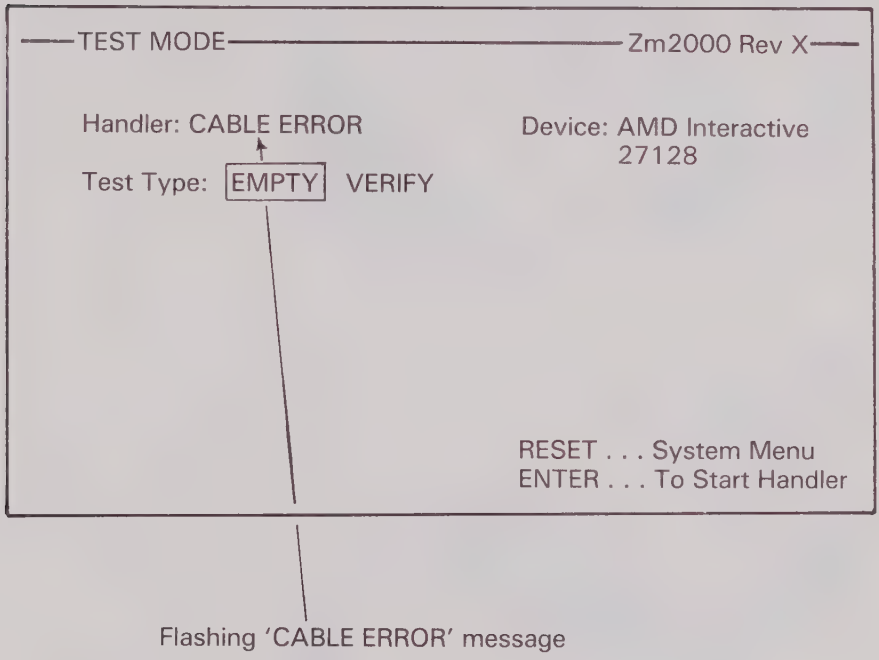


'Handler Time-out' message displayed in reverse video.

(iii) **Cable Error**

Should the handler cable become disconnected from the programmer, then the message 'CABLE ERROR' will be displayed flashing instead of the handler make and type.

For example:


TEST MODE
Zm2000 Rev X
Handler: CABLE ERROR
Device: AMD Interactive 27128
Test Type: EMPTY VERIFY
RESET . . . System Menu
ENTER . . . To Start Handler
Flashing 'CABLE ERROR' message

After the connection has been restored:

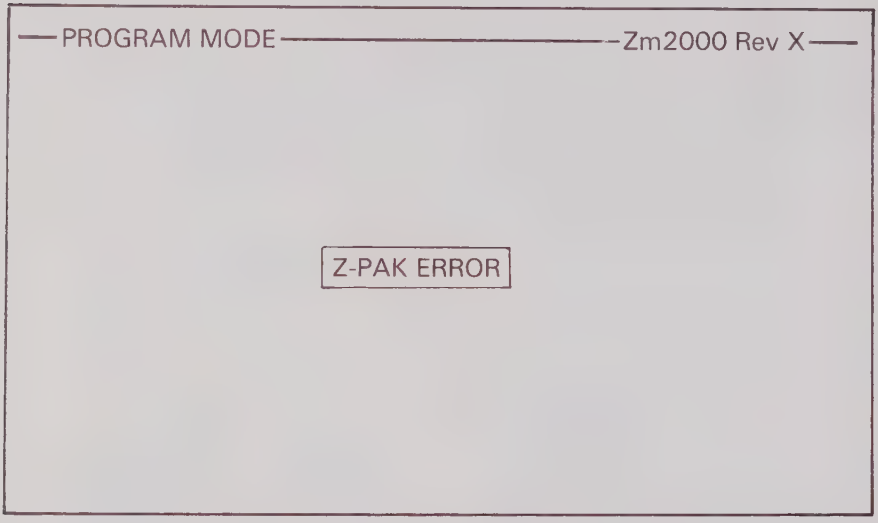
Press ENTER

To return the normal mode of operation.

(iv) **Z-PAK Error**

If the Z—Pack is dis-engaged from the socket in the pack rack and an attempt is made to enter either the Test or Program modes the message 'Z-PAK ERROR' will be displayed in the centre of the screen.

For example:

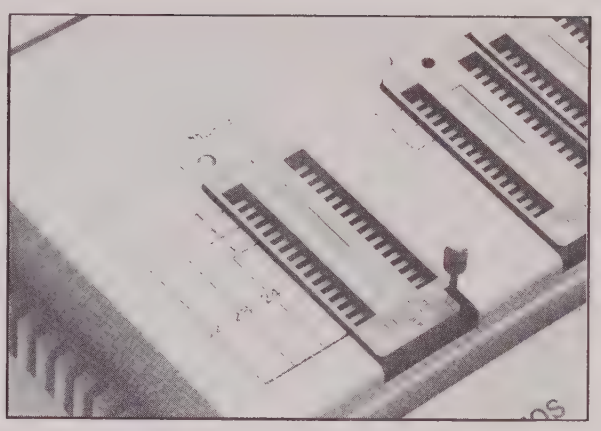


To correct this it is necessary to power-down the PPZ, re-insert the Z-pack and then power-up the PPZ again.

Correct Operation of ZIF Sockets.

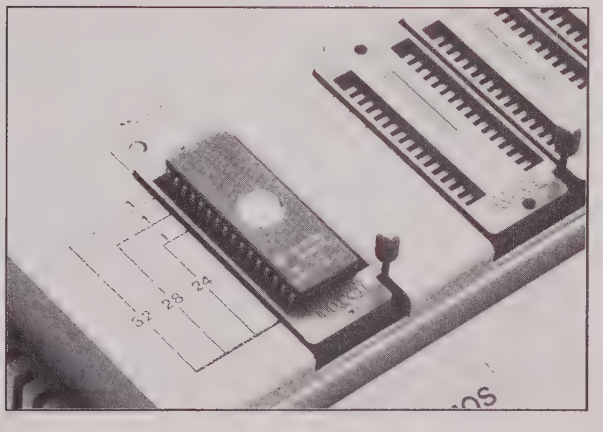
1. Empty ZIF socket.

   Lever in open (up) position.



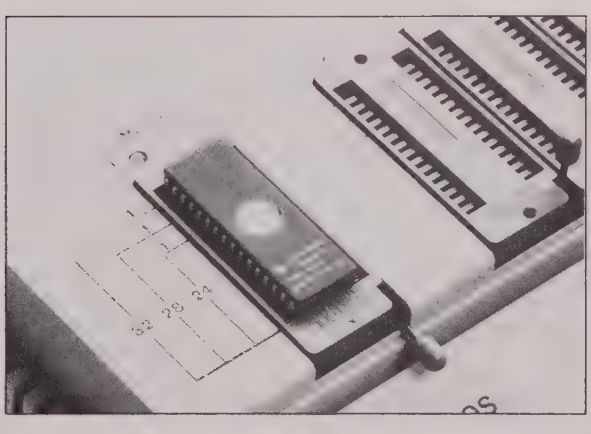
2. Device inserted.

   Socket open.



3. Device inserted.

   Lever in closed (down) position.



FREE SOFTWARE UPDATE OFFER

This Voucher entitles the owner to a FREE Software Update for the PPZ Programmer and Associated Modules within Six Months of Purchase.

Name:..... Title:.....

Company: .....

Address: .....

.....

Telephone No:..... Ext:.....

Programmer Type:..... Serial Number:.....

Module Type(s):..... Serial Number(s):.....

Date of Purchase:.....

Purchase Order Number:.....

TELL ME MORE!

ADDITIONAL TECHNICAL INFORMATION IS AVAILABLE FOR THE FOLLOWING:
(Please tick the appropriate box)

- ☐ PPZ Universal Programmer
- ☐ PP39 Portable MOS Programmer
- ☐ PP40 Ganged Copier
- ☐ PP41 Ganged Programmer
- ☐ PP42 Gang/Set Programmer
- ☐ Stag Com 1: Software communication package for PROM programmers
- ☐ Stag Com 2: Software communication package for logic programmers.
- ☐ ZL30 Logic Programmer
- ☐ ZL30A Logic Programmer
- ☐ ZL33 Ganged Logic Programmer
- ☐ CUPL: Programmable Logic Compiler
- ☐ Stag Erasers
- ☐ Please have a sales engineer call me.

Address: .....

Name: ....

Title: ....

Company: ..... Tel. No:..... Ext:.....

TELL ME MORE!

ADDITIONAL TECHNICAL INFORMATION IS AVAILABLE FOR THE FOLLOWING:
(Please tick the appropriate box)

- ☐ PPZ Universal Programmer
- ☐ PP39 Portable MOS Programmer
- ☐ PP40 Ganged Copier
- ☐ PP41 Ganged Programmer
- ☐ PP42 Gang/Set Programmer
- ☐ Stag Com 1: Software communication package for PROM programmers
- ☐ Stag Com 2: Software communication package for logic programmers.
- ☐ ZL30 Logic Programmer
- ☐ ZL30A Logic Programmer
- ☐ ZL33 Ganged Logic Programmer
- ☐ CUPL: Programmable Logic Compiler
- ☐ Stag Erasers
- ☐ Please have a sales engineer call me.

Address: .....

Name: ....

Title: ....

Company: ..... Tel. No:..... Ext:.....

AFFIX
STAMP
HERE

STAG ELECTRONIC DESIGNS LTD.
STAG HOUSE
TEWIN COURT
WELWYN GARDEN CITY
HERTS. AL7 1AU

AFFIX
STAMP
HERE

STAG ELECTRONIC DESIGNS LTD.
STAG HOUSE
TEWIN COURT
WELWYN GARDEN CITY
HERTS. AL7 1AU

AFFIX
STAMP
HERE

STAG ELECTRONIC DESIGNS LTD.
STAG HOUSE
TEWIN COURT
WELWYN GARDEN CITY
HERTS. AL7 1AU

GUARANTEE/REGISTRATION: Please return within 14 days.

Name:..... Title:.....

Address:.....

.....

.....

Telephone Number:..... Ext:.....

Programmer Type:..... Serial Number:.....

Module Type(s):..... Serial Number(s):.....

Date Received:.....

GUARANTEE: ONE YEAR

Stag Electronic Designs Ltd. guarantees that all equipment sold pursuant to any resultant agreement shall be free from Defects in Material or Workmanship at the time of delivery. The buyer must provide notice to Stag within this prescribed Guarantee Period of any Defect; if the Defect is not the result of Improper Usage, Service, Maintenance or Installation and equipment has not been otherwise damaged or modified after delivery, Stag shall either replace or repair the defective part or parts of equipment or replace the equipment at Stag's option, after return of such equipment by buyer to Stag. Cost of shipment to and from Stag's facility shall be borne on account of the buyer.

MAINTENANCE SERVICE

An annual maintenance service is available outside the guarantee period of one year. For further information and a quotation, please fill in and return this voucher.

Name:..... Title:.....

Address:.....

.....

Telephone No:..... Ext:.....

Programmer Type:..... Serial Number:.....

Module Type(s):..... Serial Number(s):.....

Date Received:.....

AFFIX
STAMP
HERE

STAG ELECTRONIC DESIGNS LTD.
STAG HOUSE
TEWIN COURT
WELWYN GARDEN CITY
HERTS. AL7 1AU
UNITED KINGDOM

AFFIX
STAMP
HERE

STAG ELECTRONIC DESIGNS LTD.
STAG HOUSE
TEWIN COURT
WELWYN GARDEN CITY
HERTS. AL7 1AU
UNITED KINGDOM

MP MASTER
MA24041